Universidad Autónoma de Madrid
Escuela Politécnica Superior
Departamento de Ingeniería Informática

Aplicación de las redes bayesianas dinámicas a la predicción de series de datos y a la detección de anomalías

Trabajo de Fin de Máster presentado para la obtención del título
Máster en Ingeniería Informática y de Telecomunicaciones
(*major* en Inteligencia Computacional)

Autor
Jaime Reguero Álvarez

Directora
Julia Díaz García

Madrid, Septiembre 2011

Resumen

Este trabajo se basa en el estudio de las redes bayesianas dinámicas aplicadas a la detección de anomalías y a la predicción en series de datos. En el capítulo 1 se da una breve noción general sobre las redes bayesianas: sus inicios, sus tipos y sus usos. Entre los diferentes tipos de redes bayesianas existentes en el estado del arte, se especifica el caso de las híbridas en el capítulo 2, interesantes por su capacidad de operar con cualquier tipo de dato, tanto cualitativo como cuantitativo. En este capítulo se describe la teoría de las gaussianas condicionales, que nos permiten tener un modelo en el que se combinen variables discretas y continuas sin necesidad de llevar a cabo la discretización de éstas últimas.

Una vez aclarados los conceptos básicos, en el capítulo 3 se incorpora la noción de "tiempo a la red". Cada instante será representado por una instancia de la red y entre dichas instancias se realizarán las conexiones que representan el flujo temporal de la información. De esta forma se añade información de estados anteriores en el tiempo al modelo. Una vez definida la estructura general de la red se introducen los algoritmos de entrenamiento utilizados para llevar a cabo la inferencia sobre la estructura propuesta. El más relevante es una modificación del algoritmo "Forward" propuesto por Rabiner [13], que nos permitirá calcular la probabilidad de los diferentes estados de las variables ocultas dada una secuencia temporal de datos.

En el capítulo 4 se aplican las redes bayesianas híbridas y dinámicas, estudiadas durante el trabajo, a la creación de un modelo para realizar el control de la medición de temperatura en estaciones del bosque Andrews Forest de Oregon (USA). El fin de este experimento es validar la implementación realizada de la teoría introducida con los experimentos realizados por Dereszynski en [7]. El modelo creado se usa para detectar anomalías en los datos recibidos de los sensores de temperatura. Esta detección de anomalías se llevará a cabo mediante la comparación de los datos obtenidos en tiempo real y la salida de un modelo predictivo de temperatura. Al ser el sensor una variable oculta de la red, es necesario utilizar la modificación del algoritmo "Forward" introducida en el capítulo 3 para calcular las probabilidades de los distinto estados que puede tomar el sensor.

Otro experimento llevado a cabo es el de crear un nuevo modelo predictivo, basado en las redes bayesianas híbridas y dinámicas explicadas en los capítulos teóricos, para mejorar las predicciones de radiación solar a partir de medidas reales de radiación solar. Se hará uso de las medidas reales de radiación obtenidas en el aeropuerto de Granada y como predicción de radiación solar de referencia se usará la realizada por el centro de meteorología "European Center for Medium-Range Weather Forecasts" (ECMWF).

Agradecimientos

En primer lugar, quiero agradecer a mi tutora Julia Díaz la guía y el apoyo que me ha brindado a lo largo del Máster. También quiero agradecer a José Dorronsoro la experiencia aportada y el tiempo dedicado a la realización de este trabajo.

Que menos que agradecer a mis padres el esfuerzo por darme una buena educación, así como su apoyo incondicional en mis decisiones. También agradecer a Irene su ayuda y apoyo, además de su paciencia $\rightarrow \infty$.

Agradecer también a Red Eléctrica de España la prestación de datos para realizar este trabajo.

Por último agradecer el apoyo económico de la beca de Máster del Instituto de Ingeniería del Conocimiento.

Índice general

Capítulo 1

Introducción a las Redes Bayesianas

En este capítulo se introducen, en primer lugar, unos conceptos básicos sobre grafos que se utilizarán a lo largo del trabajo. A continuación, se realiza una breve introducción a las redes bayesianas como base a la teoría a desarrollar en el siguiente capítulo.

1.1. Conceptos básicos sobre grafos

En primer lugar, basándonos en el trabajo de Stephenson [14], se van a introducir una serie de conceptos básicos sobre grafos que serán necesarios a lo largo del trabajo. Un **grafo** es un conjunto de **nodos** (o vértices) y un conjunto de **aristas** (o arcos), estando cada arista representada por dos nodos. Un **grafo no dirigido** es aquél en el que las aristas se pueden recorrer en cualquier dirección. Por otra parte, si los nodos que representan la arista están ordenados, entonces la arista tiene una única dirección en la que puede ser recorrida y, por tanto, se tiene un **grafo dirigido**.

Una **cadena** es una serie de nodos donde cada nodo en la cadena está conectado al anterior por una arista. Un **camino** es una cadena en la que cada arista tiene una dirección, llevando ésta la misma dirección que la cadena. Un **camino simple** es un camino con nodos únicos. Un **ciclo simple** es un ciclo donde todos los nodos son únicos, salvo el nodo de inicio/fin. Un **grafo dirigido y acíclico**, de las siglas en inglés **DAG** (***Directed Acyclic Graph***), es un grafo dirigido que no tiene ningún ciclo.

Una relación **padre/hijo** en un grafo dirigido se da cuando existe una arista, (X_1, X_2), de X_1 a X_2. En ese caso X_1 es denominado padre de X_2 y X_2 hijo de X_1. En el caso de los grafos no dirigidos, dos nodos conectados por una arista se definen como **vecinos**.

Un **grafo moral** es construido a partir de un DAG. Para ello se tiene que añadir una arista no dirigida entre cada pareja de padres de un nodo. Después, se debe quitar la

direccionalidad a las aristas iniciales obteniéndose un grafo no dirigido.

Para un camino o ciclo dado, una **cuerda** es una arista que no aparece en el camino pero que une dos nodos del camino.

El término **triangulado** describe un grafo no dirigido donde cualquier ciclo simple con al menos 4 nodos tiene al menos una cuerda. En el capítulo 2 se verá un algoritmo utilizado para triangular grafos.

El término **completo** describe un grafo no dirigido donde cada nodo esta conectado con todos los demás nodos.

Un **clique** de un grafo es un subgrafo completo que no puede ampliarse manteniendo la condición de completo.

Un **árbol de unión** es un árbol en el que los nodos son los cliques de un DAG que ha sido moralizado y triangulado.

1.2. Redes bayesianas

Una vez introducidos los conceptos básicos a usar sobre grafos se va a realizar una breve introducción a las redes bayesianas.

Un modelo gráfico es un grafo, G, con nodos, V, y aristas, E, y un conjunto P de funciones de distribución de probabilidad. Una red bayesiana es un tipo específico de modelo gráfico que es representado como un grafo dirigido y acíclico. Los nodos en un DAG, habitualmente denominados variables, son representaciones gráficas de objetos y eventos del mundo real. En el caso de las redes bayesianas son variables aleatorias. Relaciones causales entre nodos son representadas por una arista entre ellos. La arista será dirigida, empezando en la variable causal y terminando en la variable efecto. En el caso de las redes bayesianas, la ausencia de arista representa la independencia condicional entre los nodos. Además, cada nodo en el DAG tiene asociada una función de distribución de probabilidad, cuya dimensión y definición depende de las aristas que llegan al nodo. En la figura 1.1 podemos ver un ejemplo sencillo de red bayesiana.

Las redes bayesianas representan explícitamente nuestro conocimiento sobre los elementos en el sistema y las relaciones que existen entre ellos. Estas relaciones operan propagando conocimiento a través de la red una vez se tiene evidencia sobre alguno de

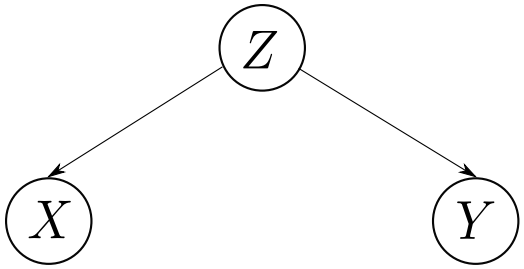


Figura 1.1: Ejemplo de red bayesiana sencilla. Las variables aleatorias X, Y, Z son los nodos del grafo y las aristas representan relaciones causales entre dichas variables. En este caso, las variables aleatorias X, Y tienen una dependencia causal con la variable Z.

los objetos o eventos del sistema. De esta manera, se pueden "aprender" las probabilidades de todos los elementos de la red a partir del conocimiento de algunos de ellos y de las relaciones condicionales entre ellos.

Las redes bayesianas tienen una importante propiedad: del grafo se puede inferir fácilmente la distribución de probabilidad conjunta para todas las variables usando la regla de la cadena para expresar la probabilidad conjunta como el producto de las probabilidades condicionales. Por tanto, la probabilidad conjunta de las n variables aleatorias $X_1, X_2, \ldots, X_n$, representada por $P(X_1, X_2, \ldots, X_n)$, se calcula como:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i | Pa(X_i)), \tag{1.1}$$

donde $Pa(X_i)$ es el conjunto de los padres del nodo X_i. Aquellos nodos que no tengan padres se denominarán nodos raíz y será necesario conocer su probabilidad a priori, $P(X_i)$.

Se denomina **marginalización** al proceso de determinar la probabilidad marginal de cada nodo, para cada uno de sus estados, en función de las probabilidades condicionales, de la estructura de la red y de la distribución de probabilidad conjunta. Por ejemplo, para calcular la probabilidad marginal de $X_k = k$ se tiene,

$$P(X_k = k) = \sum_{i_1} \cdots \sum_{i_{k-1}} \sum_{i_{k+1}} \cdots \sum_{i_n} P(X_1 = i_1, \ldots, X_{k-1} = i_{k-1}, X_k = k, X_{k+1} = i_{k+1}, \ldots, X_n = i_n)$$

donde $P(X_1 = i_1, \ldots, X_{k-1} = i_{k-1}, X_k = k, X_{k+1} = i_{k+1}, \ldots, X_n = i_n)$ se calcula como en la ecuación 1.1.

Cuando introducimos un conocimiento en la red, es decir, cuando modificamos alguna de las probabilidad marginales para que tome unos valores concretos, esta información

se va propagando a través de la red actualizando en cada paso las probabilidades de los nodos vecinos. En redes sencillas las probabilidades marginales de cada estado se pueden calcular a partir del conocimiento de la distribución conjunta usando la regla del producto y el teorema de Bayes, como se ve en [15]. Por ejemplo, para la red de la figura 1.1 supongamos que los estados de X son $(x_1, \ldots, x_p)$ y que Z puede tomar los estados $(z_1, \ldots, z_r)$, entonces la probabilidad de que X tome el valor x_l se calcularía de la siguiente forma:

$$P(X = x_l) = \sum_{i=1}^{r} P(X = x_l | Z = z_i) P(Z = z_i) = \sum_{i=1}^{r} P(X = x_l, Z = z_i)$$

Resumiendo, para construir una red bayesiana se debe especificar la distribución de probabilidad condicional para cada nodo X_i que tenga padres, $P(X_i|Pa(X_i))$, así como la probabilidad a priori, $P(X_i)$, de los nodos raíz.

1.2.1. Inferencia

Se denomina **inferencia** a la acción de calcular la probabilidad de cada estado de un nodo en una red bayesiana cuando se conocen los valores que toman otras variables de la red. Para realizar inferencia en la red es necesario estudiar primero cómo se propaga el conocimiento en la red. Esto es, dadas las observaciones de otras variables, cómo se actualizan las distribuciones del resto de las variables de la red.

Las variables de una red bayesiana, representadas por V, pueden dividirse en dos grupos dependiendo de su capacidad de ser observadas. Consideremos la partición $V = Q \cup O$. Sean $Q = \{q_0, q_1, \ldots, q_{N-1}\}$, $O = \{o_0, o_1, \ldots, o_{M-1}\}$. Denotamos los dos subconjuntos como los grupos de variables ocultas y observables respectivamente. Entonces, dado U subconjunto arbitrario de V, el objetivo de la inferencia es encontrar la función de distribución de probabilidad condicionada de que U tome el valor $\vec{u} = \{u_1, u_2, \ldots, u_K\}$ dadas las variables observadas O. Esto se puede escribir como $P(U = \vec{u}|O)$.

Si $U \subseteq O$, se define la función de distribución de probabilidad como $P(U = \vec{u}|O) = \prod_{k=1}^{K} \delta(u_k - o_k)$, donde o_k es el valor observado para la variable k-ésima del conjunto U. Siendo $\delta(x) = 1$ si $x = 0$ y $\delta(x) = 0$ en cualquier otro caso.

Si $U \subseteq Q$, se presenta un caso no trivial. Ahora se puede obtener la función deseada usando la regla de Bayes

$$P(U = \vec{u}|O) = \frac{P(U = \vec{u}, O)}{P(O)} = \frac{P(U = \vec{u}, O)}{\sum_{\forall \vec{u}} P(U = \vec{u}, O)}. \tag{1.2}$$

Se observa que es suficiente con calcular la función conjunta $P(U = \vec{u}, O)$ y marginalizar sobre U. Además, la probabilidad conjunta de U y O se obtiene marginalizando $P(V)$ sobre el conjunto de las variables ocultas $Q \setminus U$, como se indica en la siguiente expresión:

$$P(U = \vec{u}, O) = \sum_{x \in Q \setminus U} P(x, U = \vec{u}, O). \tag{1.3}$$

La inferencia en redes bayesianas arbitrarias es, en general, un problema NP completo [4]. Aún así, existen topologías de redes bayesianas, como los poliárboles que son grafos en los que cualquier par de nodos están unidos por un único camino, que permiten algoritmos de inferencia eficientes. La inferencia puede realizarse por:

1. Propagación exacta de probabilidades en una red simplemente conectada, que es aquella en la que sólo existe un camino entre dos variables de la red.

2. Inferencia aproximada (mediante técnicas de inferencia por Monte-Carlo [12], inferencia por simulación ancestral [8], etc.)

1.2.2. Aprendizaje

La fase de aprendizaje de una red bayesiana consiste en ajustar sus parámetros de forma que las funciones de distribución de probabilidad definidas por la red describan el comportamiento estadístico de los datos observados.

Un método muy utilizado para encontrar los parámetros más adecuados de una función de distribución a partir de datos observados es el método de máxima verosimilitud. En este método se parte de unos datos observados, $X = \{x_1, \ldots, x_n\}$, e independientes sobre los que se supone que vienen de una función de distribución desconocida con función de densidad $f_X(.)$. Se sabe que f_X pertenece a una familia de distribuciones $\{f(.|\theta), \theta \in \Theta\}$, denominada modelo paramétrico, de manera que f_X se corresponde con $\theta = \theta_X$, siendo θ_X el valor de los parámetros que hacen que la función de distribución genere los datos de los que partimos. Se desea encontrar el valor $\hat{\theta}$ que esté lo más próximo posible a θ_X.

Se parte de la función de densidad conjunta de todas las observaciones,

$$f(x_1, x_2, \ldots, x_n|\theta) = f(x_1|\theta) \cdot f(x_2|\theta) \ldots f(x_n|\theta). \tag{1.4}$$

Esta función de densidad se puede expresar de manera que los valores observados sean fijos, puesto que son datos, y θ sea variable:

$$\pounds(\theta|x_1, x_2, \ldots, x_n) = \prod_{i=1}^{n} f(x_i|\theta). \tag{1.5}$$

A esta función se le denomina **función de verosimilitud**. En la práctica se suele usar el logaritmo de dicha función, transformando así las multiplicaciones en sumas:

$$\hat{\ell}\left(\theta|x_1, x_2, \ldots, x_n\right) = \frac{1}{n}\log \pounds = \frac{1}{n}\sum_{i=1}^{n}\log\left(f\left(x_i|\theta\right)\right). \tag{1.6}$$

El método de máxima verosimilitud estima θ_X buscando el valor que maximiza $\hat{\ell}\left(\theta|X\right)$:

$$\hat{\theta} = \arg\max_{\theta\in\Theta}\hat{\ell}\left(\theta|x_1, x_2, \ldots, x_n\right) \tag{1.7}$$

En ocasiones el parámetro $\hat{\theta}$ se puede obtener analíticamente de los datos, aunque hay otros casos en los que hay que recurrir a optimizaciones numéricas. En este trabajo, como veremos en el capítulo de experimentos 4, al hacer uso de distribuciones gaussianas los parámetros de las distribuciones, como son la media y la varianza, se obtienen directamente de los datos.

Capítulo 2

Redes bayesianas híbridas

2.1. Introducción

Las redes bayesianas híbridas son un caso particular de las redes bayesianas en el que las variables aleatorias no toman únicamente valores discretos o continuos, sino que existen variables de ambos tipos. De esta manera se puede trabajar conjuntamente dentro del modelo con datos cuantitativos y cualitativos. Por lo general, no es posible realizar inferencia exacta cuando se usa cualquier tipo de distribuciones de probabilidad condicional, siendo necesario recurrir a algún tipo de aproximación.

Una primera opción para paliar este problema es discretizar los valores continuos y trabajar con una red bayesiana en la que todos sus estados son discretos. Otra opción, propuesta por Lauritzen [10] para los modelos híbridos, es asumir que la distribución condicional de las variables continuas dadas las discretas sea una Gaussiana multivariante. Esta última aproximación es la que se describe en este capítulo y la que se usa en esta memoria. Una explicación más detallada se puede encontrar en [5].

A lo largo de este capítulo se van a seguir los siguiente pasos: en primer lugar se describirán las distribuciones gaussianas condicionales propuestas por Lauritzen [10]. Una vez definida la teoría probabilística sobre la que se va a trabajar, se describirá el proceso de convertir el grafo dirigido y acíclico que representa la red en un árbol de unión para realizar inferencia sobre la red. Por último, se describirán las operaciones a realizar sobre el árbol para llevar a cabo la inferencia.

2.2. Distribuciones Condicionales Gaussianas

Sea G un grafo que tiene un conjunto de vértices, V, y un conjunto de aristas, E. Los vértices representan las variables de la red, mientras que las aristas indican las dependencias condicionales entre variables, es decir, cómo influyen unas en otras.

Al tratarse de una red híbrida, el conjunto de variables, V, se divide en dos conjuntos, $V = \Delta \cup \Gamma$, donde Δ representa las variables discretas y Γ representa las variables continuas. De esta forma, un elemento del espacio conjunto sería:

$$x_\rho = (i_\delta, y_\gamma)$$

con $\rho \in V$, $\delta \in \Delta$ y $\gamma \in \Gamma$. La distribución conjunta de las variables tiene como función de densidad:

$$f(x) = f(i, y) = \chi(i) \exp\left\{ g(i) + h(i)' y - \frac{y' K(i) y}{2} \right\} \tag{2.1}$$

donde $\chi(i) \in \{0, 1\}$ toma el valor 1 cuando la función f es positiva en i y x' representa la traspuesta de x. Las funciones (g, h, K), donde g es un valor real, h es un vector de dimensión $\dim(\Gamma) = p$, y K es una matriz de dimensión $p \times p$, son las **características canónicas** de la distribución y sólo están definidas cuando $\chi(i) = 1$.

Una variable, X, sigue una **distribución Condicional Gaussiana (CG)** si la distribución condicional de las variables continuas dadas las discretas sigue una distribución gaussiana multivariante:

$$P(X_\Gamma | X_\Delta = i) = N_{|\Gamma|}(\xi(i), \Sigma(i)) \text{ siempre que } p(i) = P\{X_\Delta = i\} > 0$$

donde $|\Gamma|$ es la dimensión de Γ. La dupla de funciones (ξ, Σ) define las **características de momento** de la distribución.

A continuación, se extiende la noción de distribución CG a **potencial CG**. Un potencial CG es cualquier función ϕ de la forma:

$$\phi(x) = \phi(i, y) = \chi(i) \exp\left\{ g(i) + h(i)' y - \frac{y' K(i) y}{2} \right\}. \tag{2.2}$$

Antes, al tratarse $f(x)$ de una función de densidad, K era definida positiva y simétrica. Ahora solamente se asume que $K(i)$ es simétrica. Al no ser K definida positiva, se debe tener en cuenta que ϕ no es necesariamente un densidad y, aunque se sigue usando

(g, h, K) como características canónicas de ϕ, éstas sólo estarán bien definidas cuando K sea definida positiva para todo i tal que $\chi(i) = 1$. En ese caso, se escribe ϕ como la densidad gaussiana, $\exp\left\{\frac{(y-\xi(i))'\Sigma(i)^{-1}(y-\xi(i))}{2}\right\}$, donde ξ y Σ se pueden calcular a partir de las características canónicas:

$$\xi(i) = K(i)^{-1} h(i) \tag{2.3}$$

$$\Sigma(i) = K(i)^{-1} \tag{2.4}$$

También se puede realizar la transformación en el otro sentido: partiendo de la función de probabilidad, p, y de las características de momento, (ξ, Σ), se pueden obtener las características canónicas, (g, h, K) y escribir la función de densidad como el potencial ϕ:

$$K(i) = \Sigma(i)^{-1} \tag{2.5}$$

$$h(i) = K(i)\xi(i) \tag{2.6}$$

$$g(i) = \log(p(i)) + \frac{\left\{\log(\det K(i)) - |\Gamma|\log(2\pi) - \xi(i)' K(i)\xi(i)\right\}}{2} \tag{2.7}$$

2.2.1. Operaciones básicas con potenciales CG

Una vez introducido el concepto básico de las distribuciones CG y los potenciales CG, vamos a definir las operaciones entre potenciales CG que serán utilizadas más adelante al realizar operaciones de inferencia en la red.

La operaciones básicas son:

- La extensión, para llevar el potencial a un espacio de mayor dimensión.
- La restricción, para llevar el potencial a un espacio de menor dimensión.
- La multiplicación entre potenciales.
- La división entre potenciales.
- La marginalización de los potenciales, ya sea sobre variables continuas, discretas o ambas.

A continuación se detallan cada una de ellas.

Extensión Sea ϕ un potencial CG definido en $U = (I \times Y)$, donde I es el espacio en el que toman valores las variables discretas e Y es el espacio en el que toman valores las variables continuas. Se extiende ϕ a $\bar{\phi}$ definido en $W = \langle (I \times J), (Y \times Z) \rangle$, donde $I \times J$

es el nuevo espacio de las variables discretas e $Y \times Z$ es el nuevo espacio de las variables continuas, como:

$$\bar{\phi}(i,j,y,z) = \phi(i,y). \tag{2.8}$$

Sean (g,h,K) las características canónicas de ϕ, entonces las características canónicas de la extensión serán:

$$\bar{g}(i,j) = g(i), \tag{2.9}$$

$$\bar{h}(i) = \begin{pmatrix} h(i) \\ 0 \end{pmatrix}, \tag{2.10}$$

$$\bar{K}(i,j) = \begin{pmatrix} K(i) & 0 \\ 0 & 0 \end{pmatrix}. \tag{2.11}$$

Restricción Sea ϕ un potencial CG definido en $W = \langle (I \times J), (Y \times Z) \rangle$ y sea $(j,z) \in (J \times Z)$. Entonces, se define la restricción $\phi^{(j,z)}$ de ϕ a $(I \times Y)$ como:

$$\phi^{(j,z)}(i,y) = \phi(i,j,y,z). \tag{2.12}$$

Dadas las características canónicas de ϕ, para obtener las características de la restricción simplemente hay que seleccionar las partes correspondientes al espacio destino.

Multiplicación Sean dos potenciales CG, ϕ y ψ, definidos en U y W respectivamente. Se define la multiplicación $\phi\psi$ en $U \cup W$ como:

$$(\phi\psi)(x) = \bar{\phi}(x)\,\bar{\psi}(x) \tag{2.13}$$

donde $\bar{\phi}$ y $\bar{\psi}$ son las extensiones de ϕ y ψ al espacio $U \cup W$.

Sean (g_1,h_1,K_1) las características canónicas de ϕ y sean (g_2,h_2,K_2) las características canónicas de ψ. Entonces las características canónicas de la multiplicación son la suma de las funciones canónicas de los potenciales iniciales:

$$(g_1,h_1,K_1) \times (g_2,h_2,K_2) = (g_1+g_2, h_1+h_2, K_1+K_2) \tag{2.14}$$

División Sean ϕ y ψ dos potenciales CG definidos ambos en U. Se define la división de la forma natural teniendo cuidado al dividir por cero. Queda entonces la división $(\phi/\psi)(x)$ definida como:

$$(\phi/\psi) = \begin{cases} \phi(x)/\psi(x) & \text{si } \psi(x) \neq 0 \\ 0 & \text{en otro caso} \end{cases} \tag{2.15}$$

Sean (g_1, h_1, K_1) las características canónicas de ϕ y sean (g_2, h_2, K_2) las características canónicas de ψ. Entonces las características canónicas de la división son la resta de las funciones canónicas de los potenciales iniciales:

$$(g_1, h_1, K_1) / (g_2, h_2, K_2) = (g_1 - g_2, h_1 - h_2, K_1 - K_2) . \tag{2.16}$$

Esta operación será de utilidad cuando $K_1 - K_2$ siga siendo definida positiva; en otro caso tendremos un potencial pero no será función de densidad.

Marginales sobre las variables continuas El caso de la marginalización sobre las variables continuas se resuelve integrando sobre la variable sobre la que queremos marginalizar. Supongamos que tenemos:

$$y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \tag{2.17}$$

$$h = \begin{pmatrix} h_1 \\ h_2 \end{pmatrix} \tag{2.18}$$

$$K = \begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix} \tag{2.19}$$

con y_1 de dimensión p y y_2 de dimensión q. Si se quiere marginalizar sobre y_1, hay que integrar sobre y_1.

Lema 2.2.1 *La integral $\int \phi (i, y_1, y_2) dy_1$ es finita si y sólo si K_{11} es definida positiva. En este caso, la marginal sobre y_1 es igual a un CG potencial $\tilde{\phi}$ con las siguientes características canónicas:*

$$\tilde{g}(i) = g(i) + \frac{\left\{ p \log (2\pi) - \log \det K_{11}(i) + h_1(i)' K_{11}(i)^{-1} h_1(i) \right\}}{2}, \tag{2.20}$$

$$\tilde{h}(i) = h_2(i) - K_{21}(i) K_{11}(i)^{-1} h_1(i) , \tag{2.21}$$

$$\tilde{K}(i) = K_{22}(i) - K_{21}(i) K_{11}(i)^{-1} K_{12}(i) . \tag{2.22}$$

Marginales sobre las variables discretas Sea ϕ un potencial CG definido en $W = \langle (J \times T), (Y) \rangle$ En este apartado existes dos casos. El primero es el caso en el que h y K no dependen de t, es decir, $h(j,t) = h(j)$ e igual para K. Entonces se define la marginal de ϕ sobre t como:

$$\begin{aligned} \tilde{\phi}(j, y) &= \sum_t \phi (j, t, y) \\ &= \sum_t \chi (j,t) \exp \left\{ g(j,t) + h(j)' y - \frac{y' K(j) y}{2} \right\} \\ &= \exp \left\{ h(j)' y - \frac{y' K(j) y}{2} \right\} \sum_t \chi (j,t) \exp \{ g(j,t) \} , \end{aligned}$$

de donde se obtienen las siguientes características canónicas:

$$\tilde{g}(j) = \log \sum_{j\,:\,\chi(j,t)=1} \exp\{g(j,t)\}, \tag{2.23}$$

$$\tilde{h}(j) = h(j), \tag{2.24}$$

$$\tilde{K}(j) = K(j). \tag{2.25}$$

El segundo caso se produce cuando alguna o ambas de las características canónicas h y K dependen de j y de t. En este caso se va a trabajar sobre las características de momento. El objetivo es obtener:

$$P(J=j) = \tilde{p}(j),\ E(Y|J=j) = \tilde{\xi}(j),\ var(Y|J=j) = \tilde{\Sigma}(j).$$

Haciendo uso de las siguientes relaciones:

$$\begin{aligned} E(Y|J=j) &= E\{E(Y|(J,T))|J=j\}, \\ var(Y|J=j) &= E\{var(Y|(J,T))|J=j\} + var\{E(Y|(J,T))|J=j\}, \end{aligned}$$

se obtiene:

$$\begin{aligned} \tilde{p}(j) &= \sum_t p(j,t), \\ \tilde{\xi}(j) &= \sum_t \xi(j,t)\, p(j,t)/\tilde{p}(j), \\ \tilde{\Sigma}(j) &= \sum_t \Sigma(j,t)\, p(j,t)/\tilde{p}(j) + \sum_t \left(\xi(j,t) - \tilde{\xi}(j)\right)' \left(\xi(j,t) - \tilde{\xi}(j)\right) p(j,t)/\tilde{p}(j). \end{aligned}$$

Marginalización sobre variables discretas y continuas A la hora de marginalizar sobre ambos tipos de variables, en primer lugar se marginaliza sobre las variables continuas y en segundo lugar sobre las discretas. Diremos que la marginalización es fuerte cuando al marginalizar sobre las variables discretas tengamos que las (h, K) de las características canónicas sean independientes de t. En caso contrario diremos que tenemos una marginalización débil. Se denota la marginalización a U sobre $W \setminus U$ como $\sum_{W\setminus U} \phi_W$.

2.3. Transformación de grafo a árbol de unión

Una vez se han descrito los potenciales CG y las operaciones que se pueden realizar entre ellos, se debe transformar la estructura de la red para poder operar con las variables y transferir información de unas a otras. En este capítulo vamos a ver los diferentes pasos para realizar la conversión del grafo inicial, que representa a la red bayesiana, a un árbol de unión, cuyos nodos son los cliques del árbol. Esta transformación nos permitirá realizar la inferencia de la red sobre el árbol de unión de una forma sencilla.

Se parte de un grafo, G, dirigido y marcado, con vértices, $V = \Delta \cup \Gamma$, y aristas, E. Se recuerda que Δ es el subconjunto de los vértices asociados a variables discretas y Γ es el subconjunto de los vértices asociados a variables continuas. Para transformar el grafo a un árbol de unión seguiremos varios pasos:

1. En primer lugar se moraliza el grafo.
2. En segundo lugar se triangula el grafo.
3. Por último, se construye el árbol usando los cliques como nodos.

Para poder comprender bien los pasos de la transformación es necesario introducir unos conceptos básicos sobre descomposición de grafos, que serán desarrollados en el siguiente apartado.

2.3.1. Descomposición de grafos

En primer lugar se describen conceptos sobre descomposición de grafos. Éstos son útiles porque para poder transformar el grafo es necesario que se pueda descomponer en componentes parcialmente independientes dados por los cliques del grafo. De esta manera se pueden realizar las operaciones de forma local. A continuación se define la descomposición fuerte, de la que se hará uso más adelante en éste capítulo, y cuándo se considera que un grafo marcado y no dirigido se puede descomponer. Siendo un grafo marcado aquel grafo en el que sus nodos están marcados de alguna manera, en el caso tratado en este capítulo tenemos nodos discretos y continuos, que son las marcas de los nodos.

Descomposición fuerte: Una tripleta (A, B, C) de subconjuntos disjuntos del conjunto de vértices V de un grafo marcado y no dirigido G, se dice que es una descomposición fuerte de G si $V = A \cup B \cup C$ y se cumplen las siguientes tres condiciones:

- C separa A de B, es decir, para ir de un nodo del subconjunto A a un nodo del subconjunto B hay que pasar por el subconjunto C.
- C es completo, es decir, todos los nodos de C están conectados entre sí.
- $C \subseteq \Delta$ o $B \subseteq \Gamma$, es decir, o bien C contiene sólo nodos discretos o, en caso contrario, B contiene sólo nodos continuos.

En la siguiente sección se moralizará el grafo G dirigido y marcado para dar un grafo no dirigido y marcado. Para este último tipo de grafos, la siguiente proposición establece las condiciones necesarias para que se pueda descomponer.

Proposición 2.3.1 *Un grafo marcado y no dirigido se puede descomponer si y sólo si:*

1. *Es un grafo triangulado.*

2. *No contiene ningún camino* $(\delta_1 = \alpha_0, \ldots, \alpha_n = \delta_2)$ *entre dos vértices discretos,* δ_1 *y* δ_2*, que sólo pase por vértices continuos, a excepción de que pase por un único vértice continuo que sea vecino de ambos vértices discretos. Un ejemplo de camino prohibido se puede ver en la figura 2.1.*

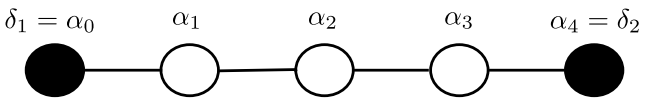


Figura 2.1: Ejemplo de camino no permitido en grafos marcados y no dirigidos que se puedan descomponer. Los círculos negros son variables discretas y los blancos continuas.

Si un grafo marcado y no dirigido cumple las condiciones de la proposición 2.3.1, entonces o bien es completo, o bien existe una descomposición (A, B, C) en subgrafos $G_{A\cup C}$ y $G_{B\cup C}$, los cuales también se pueden descomponer. [5]

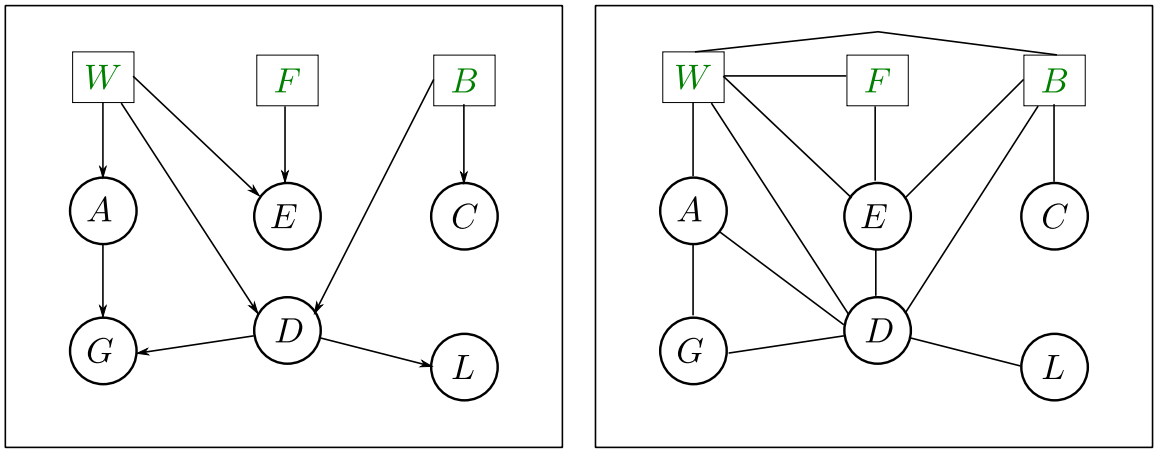
2.3.2. Moralización del grafo

En primer lugar, para poder aplicar las definiciones de descomposición vistas en el apartado anterior, se convierte el grafo G marcado y dirigido en un grafo marcado y no dirigido mediante su versión moral. Esta versión se obtiene añadiendo aristas no dirigidas entre todos los vértices que tienen hijos en común y no han sido unidos previamente, transformando a la par las aristas dirigidas en no dirigidas. Un ejemplo de la versión moral de un grafo dirigido se muestra en la figura 2.2.

2.3.3. Triangulación del grafo

Una vez se tiene un grafo marcado y no dirigido queda triangularlo para poder aplicar la proposición 2.3.1. Para realizar la triangulación se hace uso del algoritmo 1, como se propone en [5]. En dicho algoritmo se utiliza un criterio $f(v)$ para seleccionar el vértice adecuado en cada iteración. Este criterio, $f(v)$ puede definirse mediante cualquier función que asegure una buena triangulación, por ejemplo $f(v) = \#\{\text{aristas añadidas en el paso 7 del algoritmo 1}\}$ como se propone en [5].

En la figura 2.3 se muestra la evolución del algoritmo 1 sobre el grafo de la figura 2.2(b). Para cada una de las iteraciones del algoritmo, se indica en rojo la numeración actual de los vértices –la ausencia de numeración implica que el vértice está sin numerar– y en el área delimitada por la línea azul discontinua se encuentran los vértices asociados al subconjunto C_i correspondiente a la iteración actual. En este caso,



(a) (b)

Figura 2.2: Un ejemplo de grafo dirigido (a) y su grafo moral (b). En verde y cuadradas se muestran las variables discretas, en negro y redondas las continuas.

al tratarse de un grafo de 9 vértices, el contador i tomará valores $i = 1, \ldots, 9$ y representará el valor numérico a asignar a cada vértice. En cada una de las iteraciones, uno de los vértices no numerados del grafo se selecciona haciendo uso de la función $f(v) = \#\{\text{aristas añadidas en el paso 7 del algoritmo 1}\}$ y teniendo en cuenta que las variables continuas tienen preferencia sobre las discretas. Esta preferencia hace que las variables W, F y B no sean escogidas hasta que se ha completado la numeración de las variables continuas. Una vez elegido el vértice a numerar, se añaden al conjunto los vértices adyacentes que aún no han sido numerados. En la figura se marcan en cada paso de color azul aquellas aristas adyacentes al vértice recién numerado que conectan con vértices sin numerar y, por tanto, deben ser incluídos en el subconjunto. Finalmente, cada uno de los subconjuntos debe formar un grafo completo y, por tanto, la arista $B-F$ se añade al grafo en la iteración $i = 4$.

El siguiente lema describe las propiedades que posee el grafo obtenido en el algoritmo 1 y que son necesarias para la construcción del árbol de unión.

Lema 2.3.2 *Una vez completado el algoritmo 1, como salida se obtiene un grafo triangulado con una numeración perfecta* $(v_1, \ldots, v_k)$ *de sus nodos y una secuencia de eliminación asociada* $(C_1, \ldots, C_k)$ *que satisface la propiedad "running intersection".*

Se dice que $(v_1, \ldots, v_k)$ es una numeración perfecta si para cualquier nodo v_l el conjunto C_{v_l} de los vecinos de v_l, v_j, tales que $j < l$ forma un subgrafo completo.

Algoritmo 1 Algoritmo para triangular un grafo marcado y no dirigido.

1: Se comienza con todos los vértices sin numerar.
2: Se inicializa un contador $i := k$, donde k es el número de vértices.
3: **mientras** queden vértices sin numerar **hacer**
4: Se selecciona un vértice no numerado $v \in \Gamma$ que minimice el criterio $f(v)$. Si no existe tal vértice, seleccionamos un vértice no numerado $v \in \Delta$ que minimice el criterio $f(v)$.
5: Se asigna a ese vértice el número i, $v_i := v$.
6: Se crea el subconjunto C_i formado por v_i y todos sus vecinos no numerados.
7: Se añaden aristas para convertir C_i en un subgrafo completo, es decir, hasta tener una arista entre todos los pares de vértices de C_i.
8: Se elimina v_i de la lista de vértices no numerados y se decrementa el contador i en una unidad.
9: **fin mientras**

Una secuencia $(C_1, \ldots, C_k)$ es de eliminación si cada conjunto C_j cumple las siguientes propiedades:

- $v_j \in C_j$
- Los índices de cualquier nodo perteneciente a C_j distinto de v_j son menores que j.
- v_j no pertenece a ningún C_i con $i < j$.

Se dice que una secuencia $(C_1, \ldots, C_k)$ de subconjuntos tiene la propiedad *running intersection* si para todo j, $1 < j < k$, existe un $i < j$ tal que $C_j \bigcap (C_1 \cup \ldots \cup C_{j-1}) \subseteq C_i$. Los subconjuntos $(C_1, \ldots, C_k)$ representan los cliques del grafo y, en este caso, la propiedad *running intersection* significa que si cualquier variable del grafo aparece en dos subconjuntos distintos, C_l y C_m, entonces debe aparecer en todos los cliques a lo largo del camino que conecte C_l con C_m. Esta secuencia, $(C_1, \ldots, C_k)$, de subconjuntos será utilizada posteriormente para crear el árbol de unión.

2.3.4. Creación del árbol de unión

Una vez se tiene el grafo marcado, no dirigido y triangulado se puede construir el árbol de unión. Pero, debido a la mezcla de variables discretas y continuas presentes en el árbol de unión, es necesario imponer una condición más: que tenga una raíz fuerte.

Definición: Un nodo R en un árbol de unión es una **raíz fuerte** si cualquier par A,B de vecinos en el árbol, con A más cercano a R que B, satisface:

$$(B \setminus A) \subseteq \Gamma \text{ ó } (B \cap A) \subseteq \Delta, \tag{2.26}$$

entiendo la cercanía en el sentido de profundidad.

En un árbol formado por cliques, como en el que se quiere construir, la condición expresada en la definición anterior equivale a decir que $(A \setminus B, B \setminus A, A \cap B)$ forma una descomposición fuerte del subgrafo $G_{A \cup B}$, es decir, cumple las siguientes condiciones:

- $A \cap B$ separa $A \setminus B$ de $B \setminus A$,
- $A \cap B$ es completo,
- $(B \cap A) \subseteq \Delta$ ó $(B \setminus A) \subseteq \Gamma$.

Para que se cumpla el separador, $A \cap B$, debe estar compuesto unicamente de vértices discretos o, en caso contrario, los vértices del clique más lejano a la raíz y que no pertenezcan al separador, $(B \setminus A)$, deben ser continuos. Esta restricción, vista sobre el grafo que representa la red bayesiana, indica que los vértices discretos no pueden tener padres continuos.

A partir de la secuencia de subconjuntos, $C_1, \ldots, C_k$, obtenida al ejecutar el algoritmo 1 se puede construir un árbol de unión con raíz fuerte C_1. Además, la secuencia $C_1, \ldots, C_k$ obtenida en el algoritmo satisface la propiedad fuerte de *running intersection* que añade la siguiente condición a la propiedad *running intersection*:

$$C_j \setminus (C_1 \cup \ldots \cup C_{j-1}) \subseteq \Gamma \text{ ó } C_j \cap (C_1 \cup \ldots \cup C_{j-1}) \subseteq \Delta.$$

Para construir el árbol de unión a partir de la secuencia de subconjuntos, $(C_1, \ldots, C_k)$, en primer lugar hay que eliminar los conjuntos redundantes de la secuencia, es decir, aquellos subconjuntos que estén contenidos completamente en otro subconjunto. De esta manera obtenemos una nueva secuencia, $\left(\tilde{C}_1, \ldots, \tilde{C}_k\right)$, en la que todos los subconjuntos, $\tilde{C}_i$, son cliques.

Tras la ejecución del algoritmo 1 representado en la figura 2.3 se ha obtenido una secuencia de subconjuntos $(C_1 = (W), C_1 = (W, F), C_3 = (W, F, B), C_4 = (W, F, B, E), C_5 = (W, E, B, D), C_6 = (W, A, D), C_7 = (A, G, D), C_8 = (D, L), C_9 (B, C))$. Antes de construir el árbol se deben eliminar los subconjuntos redundantes. Véase que los subconjuntos C_1, C_2, C_3 y C_4 están contenidos unos en otros de la siguiente forma: $C_1 \subseteq C_2 \subseteq C_3 \subseteq C_4$, por lo que C_1, C_2, C_3 no son cliques ya que se les puede añadir una variable más y seguir siendo completos. Además la información de esos tres subconjuntos están contenida en el subconjunto C_4, por lo que se deben eliminar. La nueva secuencia, ahora de cliques, será: $\tilde{C}_1 = C_4$, $\tilde{C}_2 = C_5$, $\tilde{C}_3 = C_6$, $\tilde{C}_4 = C_7$, $\tilde{C}_5 = C_8$ y $\tilde{C}_6 = C_9$.

El siguiente paso consiste en aplicar sobre esta nueva secuencia el algoritmo 2 para obtener el árbol de unión.

Algoritmo 2 Contrucción del árbol

1: Asociar un nodo del árbol a cada clique $\tilde{C}_i$.
2: **para** i=2,. . . ,k **hacer**
3: Añadir una arista entre $\tilde{C}_i$ y $\tilde{C}_j$, donde j es cualquier valor entre 1 e $i-1$, que cumpla: $\tilde{C}_i \bigcap \left(\tilde{C}_1 \cup \ldots \cup \tilde{C}_{i-1}\right) \subseteq \tilde{C}_j$. Dicho j siempre existe, ya que la secuencia sobre la que se aplica el algortimo cumple la propiedad de *running intersection*. En dicha arista se representa la intersección entre $\tilde{C}_i$ y $\tilde{C}_j$, que será el separador.
4: **fin para**

La figura 2.4 muestra la evolución del algoritmo 2 sobre la secuencia de cliques obtenida del algoritmo representado en la figura 2.3, tras haber eliminado los subconjuntos redundantes. Inicialmente, cada uno de los cliques $\tilde{C}_1, \ldots, \tilde{C}_6$ es asignado a un nodo del árbol. Las variables continuas de cada nodo se representan en negro y las discretas en color verde. Para cada una de las iteraciones del algoritmo,$i = 2, \ldots, 6$, se marca en rojo el clique $\tilde{C}_i$ asociado a la i-ésima iteración y en azul el clique $\tilde{C}_j$ seleccionado que verifica $\tilde{C}_i \bigcap \left(\tilde{C}_1 \cup \ldots \cup \tilde{C}_{i-1}\right) \subseteq \tilde{C}_j$. Además, se marca en azul la arista añadida al árbol entre los nodos $\tilde{C}_i$ y $\tilde{C}_j$ así como los elementos de la intersección, esto es, los separadores. Es importante notar que la solución del algoritmo no es única por ejemplo, en la iteración $i = 5$, se podrían haber seleccionado como $\tilde{C}_j$ los cliques $\tilde{C}_2$, $\tilde{C}_3$ y $\tilde{C}_4$. Como criterio de selección para este ejemplo se ha escogido aquel clique con menor índice: $\tilde{C}_2$. Finalmente, observar que el grafo final es un árbol con raíz fuerte $\tilde{C}_1$.

2.3.5. Especificación del Modelo

Con la transformación del grafo al árbol se tiene la estructura sobre la que operar, que define de forma simplificada las dependencias básicas entre las variables. Se asume que la función de densidad de un vértice es el producto de las funciones de densidades condicionales de las variables asociadas al vértice dados los estados de sus padres. Aunque en principio los nodos de la red bayesiana híbrida pueden ser de cualquier tipo, para aprovechar las propiedades de los potenciales CG debemos asegurarnos que el grafo cumple la siguiente condición: si un nodo contiene variables discretas, entonces no puede tener padres que sean variables continuas. Si esta condición no se cumple, entonces no se obtendrán soluciones exactas.

Al imponer esta última condición, en el caso de las variables discretas, la densidad de dichas variables condicionada al estado de los padres se obtiene de forma sencilla, ya que los padres sólo pueden ser discretos. En el caso de que las variables sean continuas, Y, se propone tomar la distribución condicionada al estado de los padres como

$$Đ\left(Y|Pa(Y)\right) = N\left(\alpha\left(i\right) + \beta\left(i\right)' z, \gamma\left(i\right)\right),$$

donde $Pa(Y)$ es la combinación de estados discretos y continuos, (i,z), de las variables que son padres de Y, $\gamma(i)$ es un número real estrictamente positivo, $\alpha(i)$ es un número real y $\beta(i)$ es un vector de la misma dimensión que la parte continua z. Se puede apreciar que la media de la distribución depende de las variables discretas y de las continuas, mientras que la desviación sólo depende de la parte discreta de los padres.

Para calcular las características canónicas de la densidad condicional que se corresponde al potencial CG definido en la combinación de (i,z,y), vale con escribir el potencial como:

$$\phi(i,z,y) = \frac{\exp\left[\frac{-\left\{y-\alpha(i)-\beta(i)'z\right\}^2}{2\gamma(i)}\right]}{\left\{\sqrt{2\pi\gamma(i)}\right\}},$$

tomar logaritmos:

$$\log(\phi(i,z,y)) = \frac{-\left\{y-\alpha(i)-\beta(i)'z\right\}^2}{2\gamma(i)} - \frac{1}{2}\log\left(\left\{2\pi\gamma(i)\right\}\right),$$

resolver los paréntesis:

$$\frac{-\left\{yy-y\alpha(i)-y\beta(i)'z-\alpha(i)y+\alpha(i)^2+\alpha(i)\beta(i)'z-\beta(i)'zy+\beta(i)'z\alpha(i)+(\beta(i)'z)^2\right\}}{2\gamma(i)}-\frac{1}{2}\log\left(\left\{2\pi\gamma(i)\right\}\right),$$

y agrupar términos para obtener la ecuación de la forma $\phi(i,y) = \exp\left\{g(i)+h(i)'y-\frac{y'K(i)y}{2}\right\}$, quedando las características de momento (g,h,K) de la siguiente forma:

$$g(i) = -\frac{\alpha(i)^2}{2\gamma(i)} - \frac{[\log(2\pi\gamma(i))]}{2}, \tag{2.27}$$

$$h(i) = \frac{\alpha(i)}{\gamma(i)}\begin{pmatrix}1\\-\beta(i)\end{pmatrix}, \tag{2.28}$$

$$K(i) = \frac{1}{\gamma(i)}\begin{pmatrix}1 & -\beta(i)'\\-\beta(i) & \beta(i)\beta(i)'\end{pmatrix}. \tag{2.29}$$

En el ejemplo de transformación del grafo se ha visto como se pasa de un grafo acíclico, dirigido y marcado a un árbol de unión en el que, gráficamente, se ha perdido la causalidad entre las variables. Esta causalidad de padres a hijos del grafo inicial está incluida en las distribuciones de las variables y, por tanto, implícita en el modelo.

2.4. Operaciones en el árbol de unión

Una vez construido el árbol de unión y definidas las distribuciones de las variables, en esta sección se describirán las acciones a llevar a cabo para realizar inferencia sobre

el árbol de unión haciendo uso de las operaciones entre potenciales CG descritas en la sección 2.2.1.

La colección de nodos del árbol de unión se denota por C, y está formada por el conjunto de cliques del grafo. Además, S denota la colección de separadores, que hacen posible el paso de información entre cliques.

Tanto los nodos como los separadores pueden tener potenciales asociados, ϕ_W, y se asume que son potenciales CG definidos en el espacio de variables correspondiente. Se define ϕ_U, como el potencial del sistema conjunto con la siguiente expresión:

$$\phi_U = \frac{\prod_{V \in C} \phi_V}{\prod_{D \in S} \phi_D}, \tag{2.30}$$

donde V es un nodo del árbol, ϕ_V es el potencial del nodo V del árbol, D es un separador y ϕ_D es el potencial del separador D.

Al ser todos los potenciales condicionales gaussianos, la densidad conjunta del sistema será un potencial CG.

Además se asume siempre que si se tiene un nodo $V \in C$ con separador $D \in S$, entonces $\phi_D(x) = 0 \implies \phi_V(x) = 0$, lo que impide que en la ecuación 2.30 se produzca una división por 0.

2.4.1. Inicialización del árbol

Para cada nodo V, su potencial inicial ϕ_V será el producto de todos los potenciales $\phi(A)$ de los vértices asociados a él. Para los separadores se tiene que el potencial inicial es $\phi_D \equiv 1$, es decir, el potencial con características canónicas $(0, 0, 0)$.

2.4.2. Introducción de la evidencia

Las evidencias son valores concretos que toman ciertas variables y que deberán ser introducidas en todos aquellos nodos del árbol de unión que contengan dichas variables.

Existen dos tipos de evidencias:

- **Evidencia discreta:** Asigna un estado concreto a una variable discreta. Este tipo de evidencia se introduce fijando el valor de dicha variable al estado concreto.

- **Evidencia continua:** Es un valor concreto, y_γ^*, para la variable continua Y_γ. Este tipo de evidencia modifica todos los potenciales de todos los nodos del árbol de unión en los que esté contenida la variable. Se tienen que modificar de forma que la variable Y_γ esté fija al valor y_γ^*. Si el potencial tiene características canónicas (g, h, K) con:

$$h(i) = \begin{pmatrix} h_1(i) \\ h_\gamma(i) \end{pmatrix}, K(i) = \begin{pmatrix} K_{11}(i) & K_{1\gamma}(i) \\ K_{\gamma 1}(i) & K_{\gamma\gamma}(i) \end{pmatrix},$$

el potencial transformado ϕ^* tendrá características canónicas (g^*, h^*, K^*) dadas por:

$$K^*(i) = K_{11}(i), \tag{2.31}$$

$$h^*(i) = h_1(i) - y_\gamma^{*\prime} K_{\gamma 1}(i), \tag{2.32}$$

$$g^*(i) = g(i) + h_\gamma(i)' y_\gamma^* - K_{\gamma\gamma}(i) \left(y_\gamma^*\right)^2 / 2, \tag{2.33}$$

obtenidas del resultado de condicionar una de las variables en una distribución multivariante.

Una vez se ha introducido la evidencia en el árbol, el nuevo potencial representa el conocimiento condicionado a la evidencia introducida, es decir, el resto de estados de la variable discreta serán imposibles de obtenerse y la variable continua Y_γ tomará siempre el valor y_γ^*.

2.4.3. Flujo de información entre cliques

Para realizar la inferencia en la red se necesita que haya flujo de información de manera que, cuando una variable tome un valor, el resto de variables se modifiquen de acuerdo a dicho valor. Sea $V \in C$ y sea $W \in C$ un vecino de V con separador $D \in S$. Entonces el flujo de información de V a W consiste en las siguientes operaciones entre los potenciales:

$$\text{Actualización del potencial del separador } \phi_D^* = \sum_{V \setminus D} \phi_V, \tag{2.34}$$

$$\text{Paso de información del separador al potencial vecino } W \ \phi_W^* = \phi_W \frac{\phi_D^*}{\phi_D}. \tag{2.35}$$

En la primera ecuación se traspasa la información del universo de conocimiento V al separador. Esta acción se realiza marginalizando V sobre D. En la segunda ecuación se aplica el factor de actualización, es decir, la diferencia de los dos estados del separador $\frac{\phi_D^*}{\phi_D}$, al universo W mediante la multiplicación de los potenciales.

Llegados a este punto conocemos las restricciones que debemos imponer a nuestro grafo, cómo realizar la transformación del grafo a un árbol de unión sobre el que realizar la inferencia de la red y los pasos para realizar la inferencia en la red haciendo uso de las operaciones de los potenciales CG.

En el siguiente capítulo se aplicará este apartado a las redes bayesianas dinámicas y se introducirá el algoritmo "Forward" necesario para el cálculo de probabilidades a partir de una secuencia de estados.

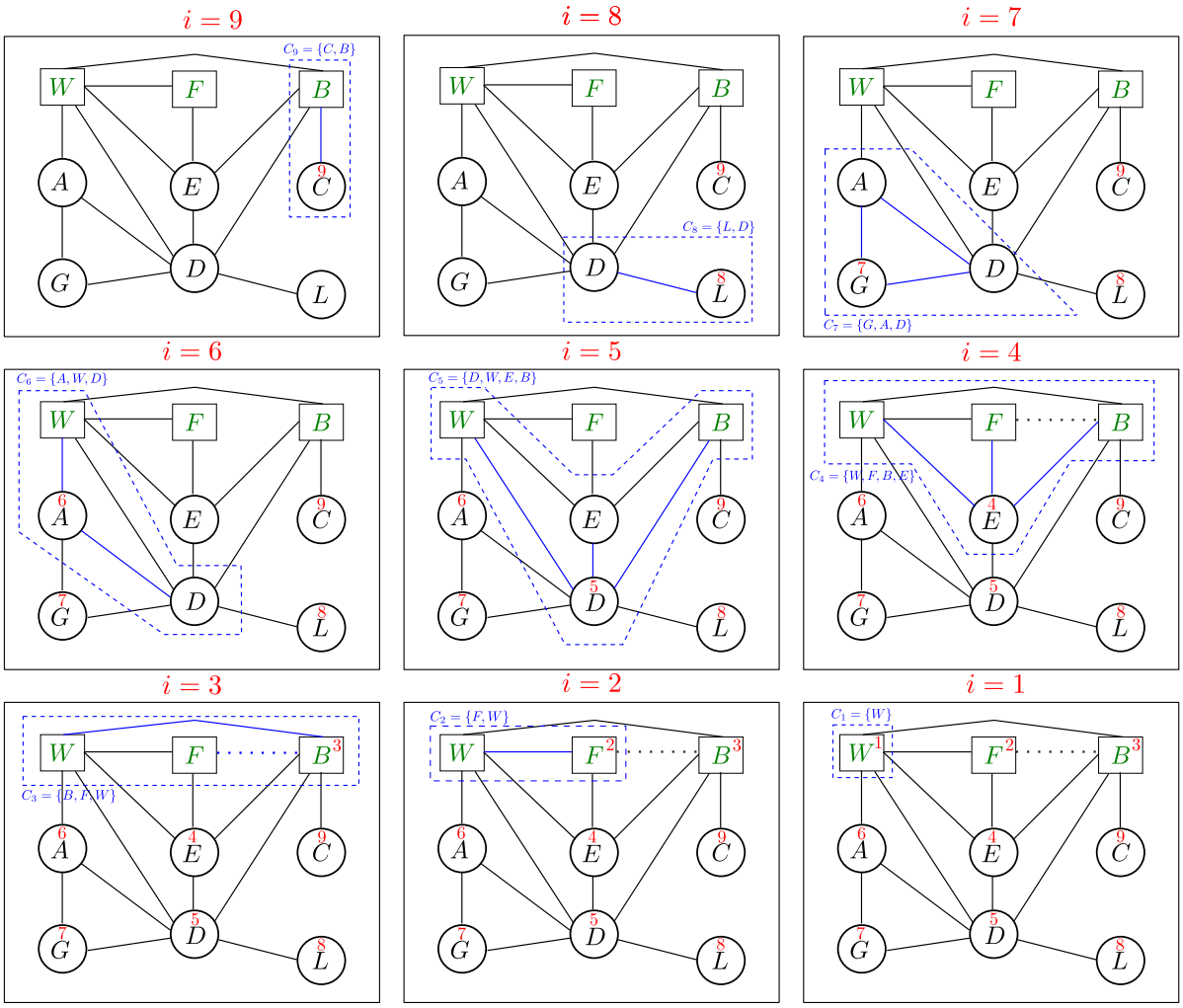


Figura 2.3: Evolución del algoritmo 1 sobre el grafo de la figura 2.2(b).

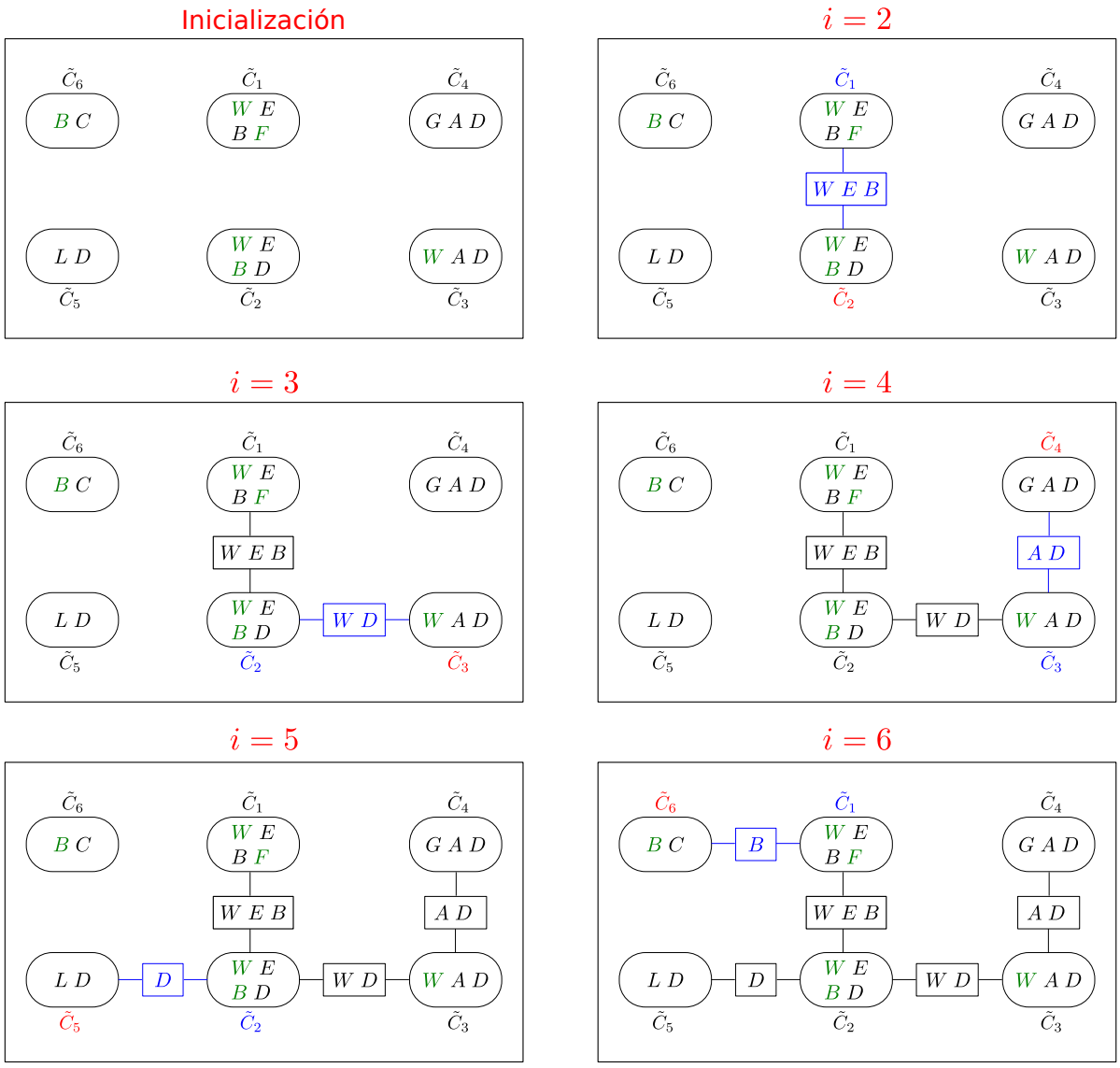


Figura 2.4: Evolución del algoritmo 2 para la construcción del árbol de unión sobre los cliques $\tilde{C}_1, \ldots, \tilde{C}_6$.

Capítulo 3

Redes bayesianas dinámicas

3.1. Introducción

En este capítulo se describe una visión general de las redes bayesianas dinámicas aplicando la teoría vista en el capítulo 2.

Una red bayesiana dinámica es una red bayesiana que representa una secuencia de variables, es decir, una red bayesiana que se va modificando a lo largo del tiempo. En la figura 3.1 se puede encontrar una forma muy general de representar las redes bayesianas dinámicas. En ella, para cada instante de tiempo, se tiene una ventana con una copia de la estructura de la red, existiendo conexiones entre las ventanas. De esta forma se tiene una red bayesiana que en cada instante de tiempo recibe información del instante anterior además de las variables observables.

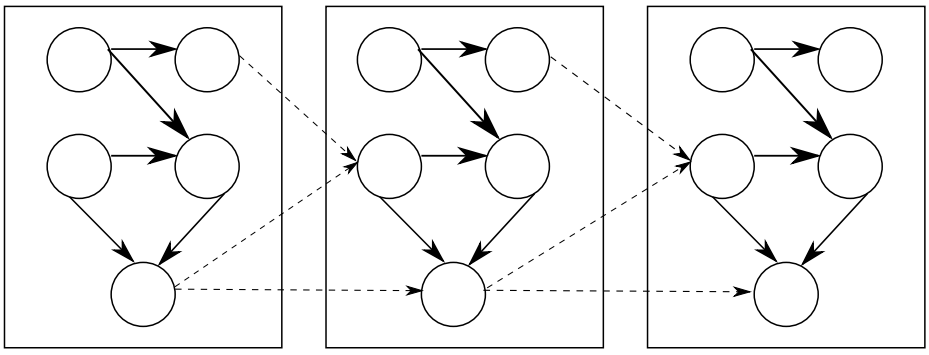


Figura 3.1: Estructura de una red bayesiana dinámica donde las flechas punteadas representan el flujo de información entre las ventanas de tiempo.

Las redes bayesianas dinámicas se han utilizado mucho para problemas de series temporales, por ejemplo en problemas de reconocimiento del habla [13]. En este caso se hace uso de un tipo de redes bayesianas dinámicas, como son los modelos ocultos de Markov, acerca de los cuales se puede profundizar en el tutorial de Rabiner [13].

3.2. Modelo general

Para poder hacer uso de la teoría descrita en el capítulo 2 es necesario que nuestra red cumpla un requisito: las variables discretas no pueden tener padres continuos; tampoco en el flujo de información entre ventanas de tiempo. En la figura 3.2, podemos encontrar un ejemplo de una red bayesiana dinámica donde aplicar la teoría de las distribuciones condicionales gaussianas. En ella se muestran las conexiones posibles entre variables, se puede ver que nunca una variable discreta, representada por cuadrados, tiene por padre una variable continua, representada por círculos.

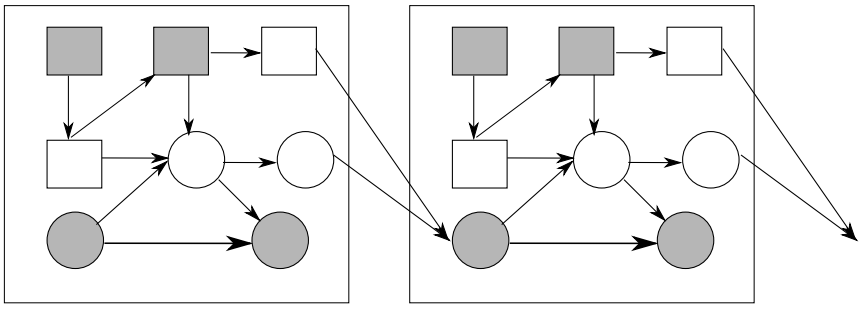


Figura 3.2: Ejemplo de red bayesiana sobre la que aplicar distribuciones condicionales gaussianas. Los círculos representan variables continuas y los cuadrados variables discretas. A su vez se consideran los elementos sombreados como variables observables.

Por otra parte, para realizar la inferencia en la red sería necesario crear un árbol de unión para cada instante de tiempo. Este árbol estará creado a partir del grafo formado por la ventana de tiempo actual y las variables de la ventana de tiempo anterior que conectan con ella y que ya son conocidas. En la práctica sólo se construye un árbol, ya que todas las ventanas tienen la misma estructura y, por tanto, la estructura del árbol permanece invariable. La componente temporal de la red se refleja entonces en el contenido de los nodos que cambia para cada instante de tiempo. En la figura 3.3 se puede ver el grafo a partir del cual se crearía la estructura del árbol de unión para la red representada en la figura 3.2.

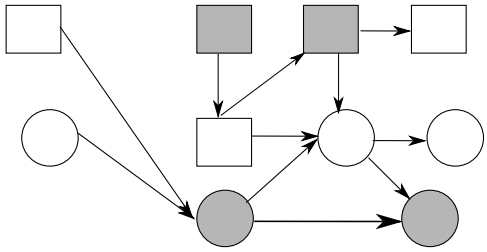


Figura 3.3: Grafo obtenido de la red bayesiana dinámica de la figura 3.2 para la construcción de árbol sobre el que realizar inferencia.

3.3. Algoritmos de inferencia

En esta sección se describe el algoritmo *Forward* [13], para propagar la información obtenida a lo largo del tiempo pasado al instante de tiempo actual.

3.3.1. Algoritmo *Forward*

Para la ejecución de la red a lo largo de su vida se propone el uso del algoritmo *Forward* propuesto por Lawrence R. Rabiner en [13]. Dada una secuencia de observaciones $O = \{O_1, O_2, \ldots, O_T\}$ y el modelo entrenado λ (consistente en el cálculo de las distribuciones de los nodos), el algoritmo consiste en calcular la probabilidad de la secuencia de observaciones dado el modelo, $P(O|\lambda)$. Aplicando directamente la teoría de la probabilidad, la solución consistiría en sumar la probabilidad conjunta de la secuencia de observaciones sobre todas las posibles secuencias de estados ocultos. Estos estados son ocultos porque son los estados que toman las variables ocultas de la red, que son aquellas que no se pueden observar pero que se pueden inferir a partir de otras variables observables. Si nuestro modelo tiene N estados ocultos, la suma de la probabilidad conjunta de la secuencia de observaciones supone un coste del orden de $2T \cdot N^T$ [13]. Rabiner propone una manera más eficiente de realizar este cálculo basándose en el algoritmo *Forward-Backward* [1, 2]. El algoritmo *Forward* consta de 3 pasos pero, en este trabajo únicamente necesitamos el primero de ellos, esto es, calcular la variable *forward*, $\alpha_t(i)$ definida como sigue,

$$\alpha_t(i) = P(O_1, O_2, \ldots, O_t, q_t = S_i|\lambda) \tag{3.1}$$

donde q_t es la variable oculta q en tiempo t. $\alpha_t(i)$ alamacena la probabilidad conjunta hasta el instante de tiempo t de la secuencia de observaciones y de la variable oculta q que toma el valor S_i en tiempo t. Los valores de α se pueden obtener de manera iterativa utilizando el algoritmo 3. En la sección de experimentos se hará uso de este algoritmo para realizar inferencia.

Algoritmo 3 Algoritmo para el cálculo iterativo de la variable α del algoritmo *Forward*.

1: Inicialización:

$$\alpha_1(j) = P(S_j)\, P(O_1|q_1 = S_j) = P(O_1, q_1 = S_j)$$

2: Inducción:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) \cdot P(q_{t+1} = S_j|q_t = S_i)\right] \cdot P(O_{t+1}|q_{t+1} = S_j)$$

El paso 1 consiste en la inicialización de la probabilidad *forward* como la probabilidad conjunta del estado S_j y la observación inicial O_1. El paso de inducción viene de:

$$\begin{aligned}
\alpha_{t+1}(j) &= P(0_1, \ldots, O_{t+1}, q_{t+1} = S_j) \\
&= (P(0_{1:t}, q_t = S_1) \cdot P(q_{t+1} = S_j|q_t = S_1) + \ldots + P(0_{1:t}, q_t = S_N) \cdot P(q_{t+1} = S_j|q_t = S_N)) \cdot P(O_{t+1}|q_{t+1} = S_j) \\
&= \left(\sum_{i=1}^{N} P(0_{1:t}, q_t = S_i) \cdot P(q_{t+1} = S_j|q_t = S_i)\right) \cdot P(O_{t+1}|q_{t+1} = S_j) \\
&= \left(\sum_{i=1}^{N} \alpha_t(i) \cdot P(q_{t+1} = S_j|q_t = S_i)\right) \cdot P(O_{t+1}|q_{t+1} = S_j)
\end{aligned}$$

Este paso esta ilustrado en la figura 3.4, en la que se muestra cómo el estado S_j puede ser calculado en tiempo $t+1$ a partir de los N posibles estados, S_i, en tiempo t. Como $\alpha_t(i)$ es la probabilidad del evento: "$O_1, O_2, \ldots, O_t$ son observados y el estado en tiempo t es S_i", entonces el producto $\alpha_t(i)\, a_{ij}$ es la probabilidad del evento conjunto en el que $O_1, O_2, \ldots, O_t$ son observados y el estado S_j es obtenido en tiempo $t+1$ a través del estado S_i en tiempo t, donde a_{ij} representa la probabilidad de transición del estado S_i al estado S_j, $P(q_{t+1} = S_j|q_t = S_i)$. Sumando el producto para los N posibles estados S_i en tiempo t se obtiene la probabilidad de S_j en tiempo $t+1$ con las observaciones previas. Una vez tenemos la probabilidad de S_j, es fácil ver que $\alpha_{t+1}(j)$ se obtiene multiplicando la suma calculada por $P(O_{t+1}|q_{t+1} = S_j)$, es decir, teniendo en cuenta la observación O_{t+1} en el estado S_j.

Obsérvese que el coste del algoritmo 3 es $O(TN)$, reduciendo significativamente el coste del cálculo por fuerza bruta.

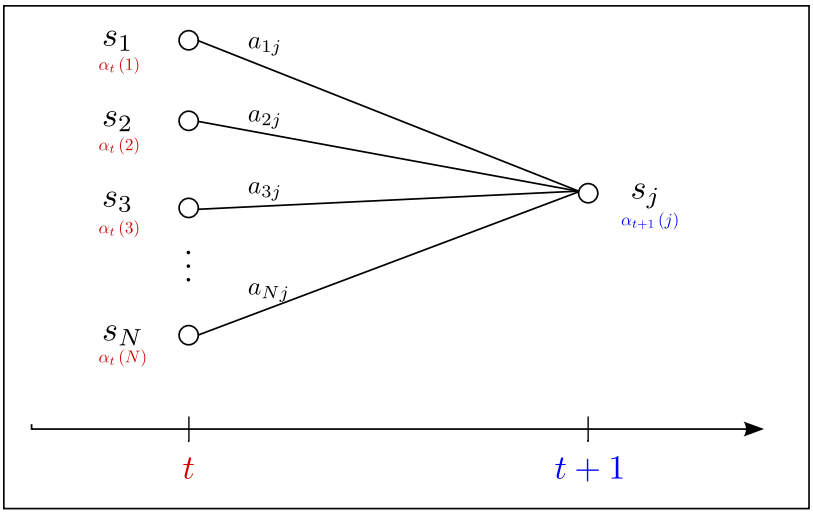


Figura 3.4: Paso de inducción del algoritmo *forward*.

Capítulo 4

Experimentos

4.1. Introducción

En esta sección se detallan los modelos propuestos para diferentes problemas, haciendo uso de la teoría descrita en los capítulos anteriores. Se proponen soluciones para dos problemas:

1. Detección de fallos en sensores de temperatura, para el que se replican los experimentos llevados a cabo por Dereszynski en [7].

2. Predicción de radiación solar incidente en la Tierra.

4.2. Detección de anomalías en sensores de temperatura

Uno de los experimentos que se han llevado a cabo para probar las redes bayesianas dinámicas usando condicionales gaussianas, es la detección de anomalías en sensores de temperatura como se describe en el artículo de Dereszynski [7]. Para ello se han usado datos del bosque *HJ Andrews Experimental Forest* [6] y la librería de código *Bayes Net Toolbox* creada por Kevin P. Murphy [11] y disponible en http://bnt.googlecode.com.

4.2.1. Problema

Este trabajo tiene como objetivo replicar los experimentos realizados por Dereszynski [7] y validar la implementación realizada haciendo uso de la librería de código Matlab *Bayes Net Toolbox*. En dichos experimentos el problema a resolver es determinar si los datos de temperatura de estaciones meteorológicas distribuidas por el bosque son correctos o existe algún fallo en el sensor. Esto es, resolver un problema de clasificación

en el que a partir de unas variables de entrada, entre las que se encuentra la observación de temperatura, se pretende etiquetar el estado del sensor. La necesidad de realizar esta clasificación existe porque:

1. Es complicado que el personal del centro de datos pueda revisar si hay indicios de anomalías en las series de temperatura dada la gran cantidad de información a manejar.

2. No siempre se puede acceder a las estaciones meteorológicas para comprobar el estado del sensor y poder determinar así la veracidad de las series de temperatura.

Por tanto, se necesita poder modelar el estado del sensor en función de la temperatura que se recibe en el centro de datos. En la siguiente sección se describe el modelo sugerido para resolver el problema.

4.2.2. Modelo

Para modelar la relación entre el sensor y la serie de temperatura se va a hacer uso de las gaussianas condicionales, propuestas por Lauritzen [10] y descritas en el capítulo 2, aplicadas a las redes bayesianas híbridas como se propone en el capítulo 3. La separación entre instantes de tiempo será de 15 minutos, que es el intervalo de refresco de las observaciones en los datos. Por tanto, la red contendrá dos variables discretas que representan el instante de tiempo en el que nos encontramos, (QH, D), donde:

1. $QH = 1, ..., 96$ representa, para un día, el cuarto de hora en el que nos encontramos.

2. $D = 1, ..., 365$ representa el día del año en el que nos encontramos.

El primer paso es realizar una predicción de la temperatura que permita decidir si la observación obtenida del sensor es aceptable o debe ser revisada/descartada. Para ello se asume que la temperatura viene dada en función de un valor base aprendido, B, y una desviación sobre el mismo, Δ. La distribución de la temperatura predicha viene por tanto dada por:

$$T \sim N\left(B_{(qh,d)} + \Delta_t, \sigma_T^2\right) \tag{4.1}$$

donde el valor base, $B_{(qh,d)}$, es la temperatura media para cada par *(día, cuarto de hora)*. Este valor se calcula en el entrenamiento del modelo aplicando medias móviles con una ventana de 3 días ($M = 3$) y 5 cuartos de hora ($N = 5$) a lo largo de 4 años ($Y = 4$):

$$B_{(qh,d)} = \frac{1}{Y\,(2M+1)\,(2N+1)} \sum_{y,u,t} T\,(d+u, qh+t, y) \tag{4.2}$$

donde
$y \in \{1, ..., Y\}$ es el índice del año.

$u \in \{-M, ..., M\}$ es el desplazamiento sobre el día.
$t \in \{-N, ..., N\}$ es el desplazamiento sobre el cuarto de hora.
En la figura 4.1 se explica gráficamente la ventana de datos seleccionada para el cálculo de la temperatura media.

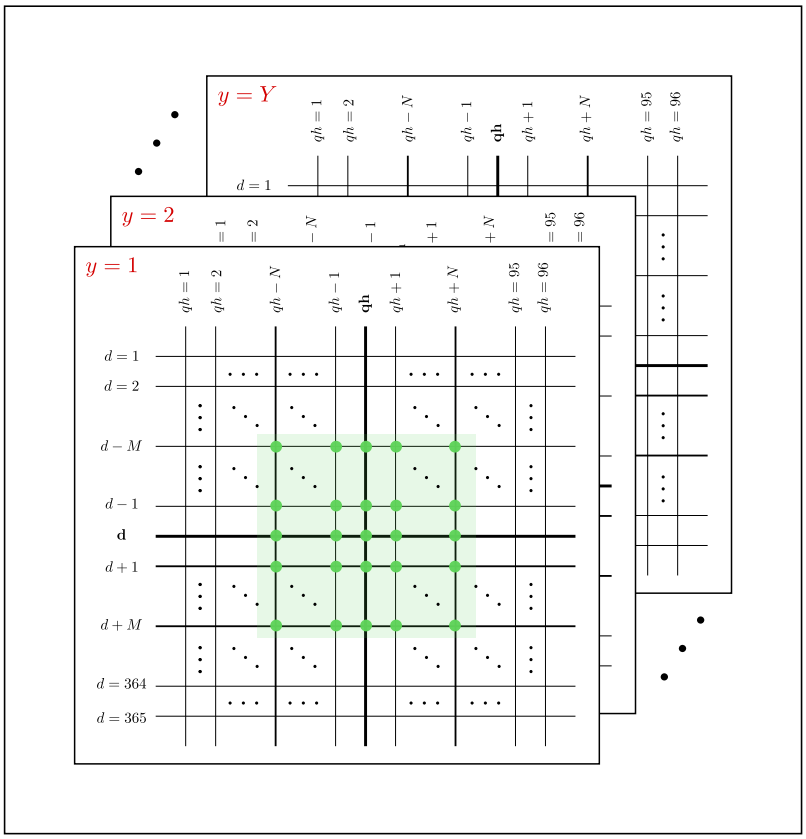


Figura 4.1: Representación de la ventana de datos seleccionada para el cálculo de $B_{(qh,d)}$.

Como comenta Dereszynski en [7], es posible que B esté sesgado. Para solventarlo propone restar un término, la primera derivada $Q\left(d, qh, u, t, y\right)$, para cada desplazamiento (u, t), eliminando así la tendencia lineal a corto plazo de la curva de la temperatura. El cálculo se representa como sigue:

$$(4.3)\, Q\left(d, qh, u, t, y\right) = \frac{1}{2M+1} \sum_{v} \left(T\left(d+u+v, qh+t, y\right) - T\left(d+u, qh, y\right)\right),$$

$$(4.4) B_{(qh,d)} = \frac{1}{Y\left(2M+1\right)\left(2N+1\right)} \sum_{y,u,t} \left(T\left(d+u, qh+t, y\right) - Q\left(d, qh, u, t, y\right)\right).$$

Esto es posible debido a que la temperatura es muy estacional, por lo que se cree que la temperatura media para un par *(día, cuarto de hora)* dado no variará mucho a lo largo de los años.

Respecto a Δ_t, representa la desviación de la temperatura sobre la base B calculada. Se puede interpretar como la tendencia local de la temperatura. Δ_t se modela como un proceso de *Markov* de primer orden con el par (qh, d) como entradas observadas. Su distribución viene dada por:

$$\Delta_t \sim N\left(\mu_{(qh,d)} + w\Delta_{t-1}, \sigma_{(qh,d)}\right), \tag{4.5}$$

en la que la media viene determinada por la media histórica de la desviación para ese par *(día, cuarto de hora)*, representado por $\mu_{(qh,d)}$, y una ponderación de la desviación en el instante de tiempo anterior, cuyo valor es $w\Delta_{t-1}$, siendo w un parámetro a seleccionar en el entrenamiento. La varianza viene dada por la varianza histórica de las desviaciones para cada par *(día, cuarto de hora)*, representada por $\sigma_{(qh,d)}$. Teniendo en cuenta la media y la varianza históricas podemos capturar más fácilmente la estacionalidad de la temperatura. De esta forma se crea la parte predictiva de la red como se aprecia en la figura 4.2.

En el entrenamiento de la red se calculan la media y la varianza históricas de la desviación usando una ventana, M, de 31 días y 4 años de datos de la siguiente forma:

$$\mu_{(qh,d)} = \frac{1}{Y\,(2M+1)} \sum_{y,u} \Delta\,(y, d+u, qh)\,. \tag{4.6}$$

$$\sigma^2_{(qh,d)} = \frac{1}{Y\,(2M+1)} \sum_{y,u} \left(\Delta\,(y, d+u, qh) - \mu_{(qh,d)}\right)^2. \tag{4.7}$$

donde $\Delta\,(y, d, qh)$ se calcula como:

$$\Delta\,(y, d, qh) = T\,(y, d, qh) - B_{(d,qh)} \tag{4.8}$$

Una vez finalizada la fase de predicción, hace falta incluir el mecanismo por el cual se realizará la estimación del estado del sensor, S, el cual se modela como una variable oculta discreta que define su estado funcional. Se definen cuatro estados según el funcionamiento: VG (Muy bueno), G (bueno), B (malo) y VB (muy malo). Nuevamente se modela S como un proceso de *Markov* de primer orden para capturar la idea de que los sensores en buen estado, o en mal estado, tienden a mantenerlo. Esta idea de que los sensores tienden a mantener su estado viene representada en la matriz de transición, en la que la probabilidad de pasar de un estado a él mismo es mucho mayor que la probabilidad de cambiar a cualquiera de los otros estados. Por otra parte, el modelo impone

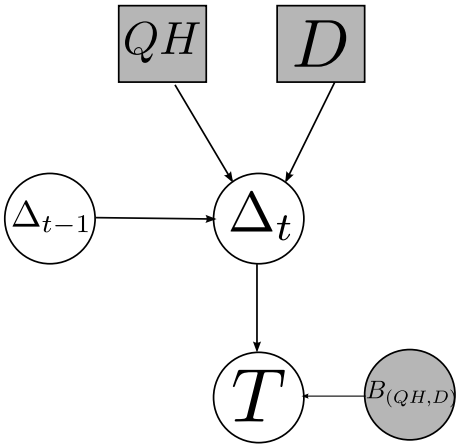


Figura 4.2: Modelo predictivo de la red. Los rectángulos representan variables discretas y los círculos las continuas. Las variables observadas están sombreadas.

que la temperatura observada, O, condicionada al estado $S = s$ siga una distribución normal :

$$O|S = s \sim N\left(\mu_s + w_s T, \sigma_s^2\right) \tag{4.9}$$

donde σ_s indica cómo está capturando la temperatura predicha, T, a la real, O, tomando valores pequeños cuando la predicción se ajusta bien a la observación y valores altos cuando la predicción se distancia mucho de la observación. En la tabla 4.1 se indican las distribuciones, impuestas como hipótesis, de la observación según el estado del sensor. En todos los casos la media sólo depende de la temperatura, T, por lo que w_s vale 1 y μ_s vale 0. Se observa que en los tres primeros estados se mantiene la media como la temperatura predicha con distintas varianzas, teniendo el estado VG muy poca varianza, la cual va aumentando según el sensor va pasando a los siguientes estados. El cuarto estado del sensor, VB, representa cuándo la temperatura no tiene ninguna relación con la predicha para el instante en el que nos encontramos.

Tras estas acciones se puede definir la red al completo, uniendo la parte predictiva al modelo de estimación del estado. La red resultante se observa en la figura 4.3.

4.2.3. Inferencia

Tras realizar el entrenamiento, en el que se calculan $B_{(qh,d)}$, $\mu_{(qh,d)}$ y $\sigma_{(qh,d)}$, como se ha especificado con anterioridad, se realiza la inferencia en la red usando los algoritmos vistos en los capítulos 2 y 3:

- *Árbol de unión*, aplicado a las CG's por Lauritzen [10].

Tabla 4.1: Distribución de la temperatura observada O en función del estado del sensor en tiempo t, S_t, y la temperatura predicha T.

ESTADO DEL SENSOR	**DISTRIBUCIÓN**	$\boldsymbol{\mu_s}$	$\mathbf{w_s}$
$O\|St = VG$	$N(T, 1.0)$	0	1
$O\|St = G$	$N(T, 5.0)$	0	1
$O\|St = B$	$N(T, 10.0)$	0	1
$O\|St = VB$	$N(0, 100000)$	0	0

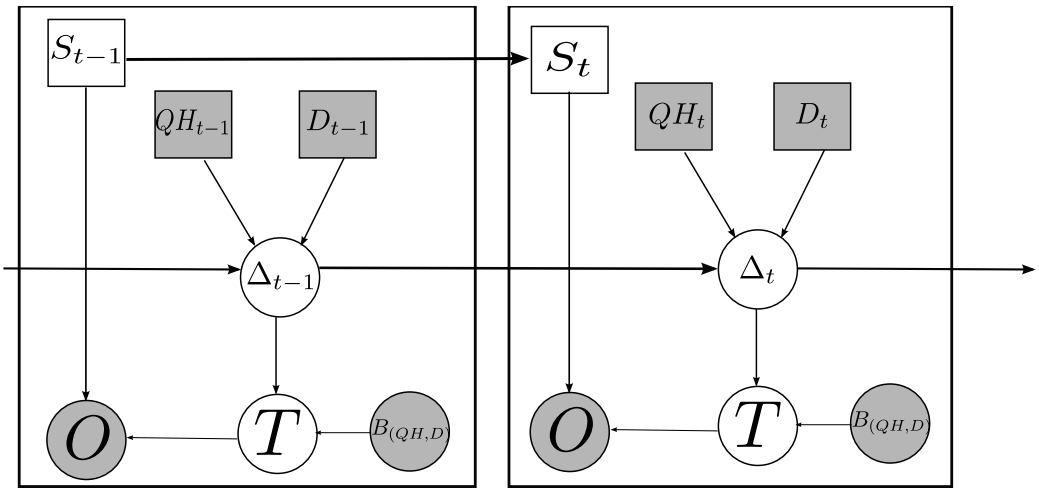


Figura 4.3: Ventana de la red en el instante de tiempo t. Los rectángulos representan variables discretas y los círculos las continuas. Las variables observadas están sombreadas.

- El cálculo de la variable α_t del algoritmo *Forward*, propuesto por Rabiner [13], y definido en el algoritmo 3.

Partiendo de la red de la figura 4.3 se deben seguir los siguientes pasos:

1. En primer lugar se obtiene el grafo que representa un instante de tiempo t, se ve el la figura 4.4.

2. En segundo lugar se crea el árbol de unión a partir del grafo del punto 1 tal y cómo se describe en el capítulo 2. La figura 4.5 representa el árbol de unión para el grafo 4.4.

3. Se realiza inferencia sobre el árbol de unión, para lo que se usa el algoritmo 4.

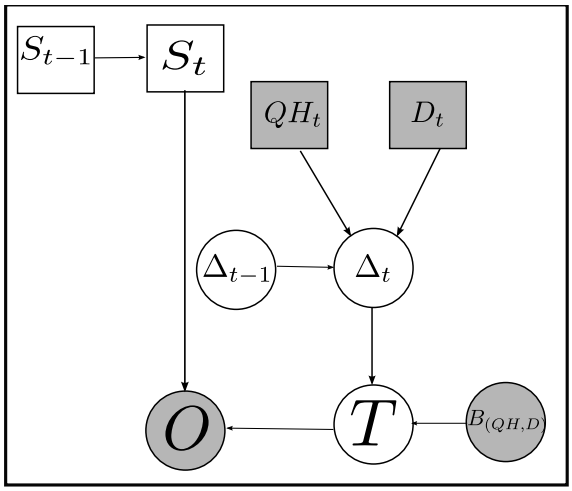


Figura 4.4: Grafo creado a partir de la ventana de tiempo t sobre el que se crea el árbol de unión.

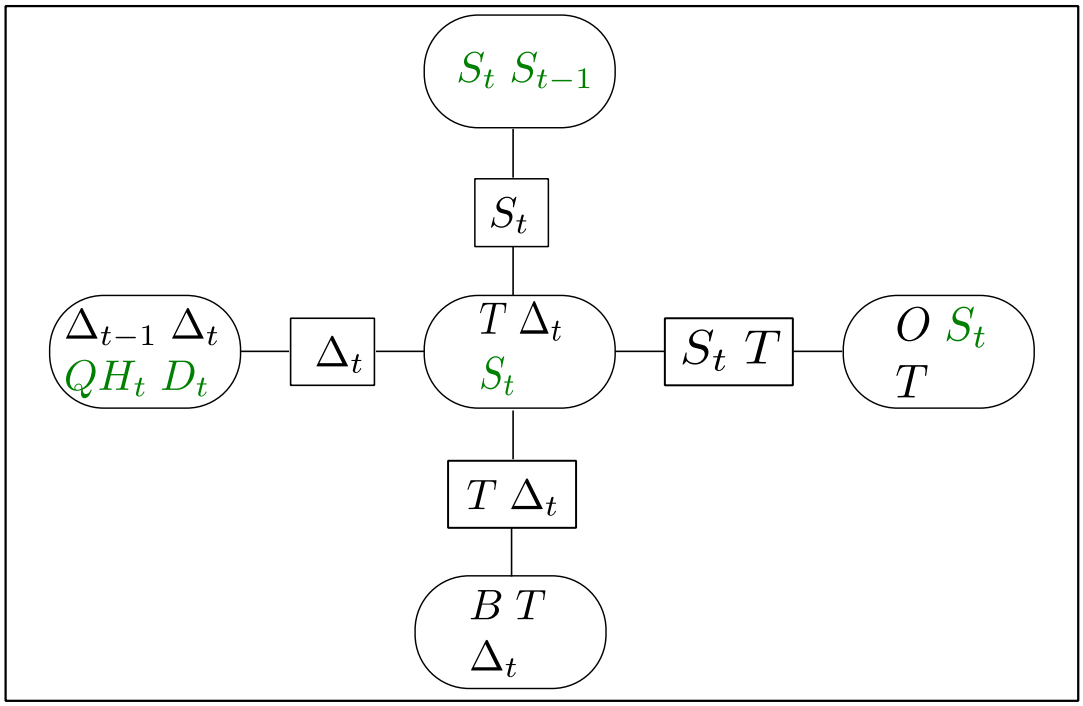


Figura 4.5: Árbol de unión generado a partir del grafo 4.4.

En primer lugar se crea el árbol de unión del grafo de la figura 4.3 para poder introducir evidencias y calcular las probabilidades necesarias. Después, se hace uso del algoritmo 4,

para llevar a cabo la inferencia en la red y así poder calcular en cada instante de tiempo el estado del sensor de temperatura.

Algoritmo 4 Algoritmo usado para la inferencia de la red.

1: Introducir el valor de las variables observadas, QH, D, B y O, en el árbol de unión.
2: Calcular la probabilidad a posteriori para S_t, P_{S_t}, haciendo uso del árbol de unión.
3: Calcular el valor más probable de S_t, $\arg\max_s P(S_t = s|O_{1:t})$. Asignarlo como valor del sensor para el dato en tiempo t y añadirlo como evidencia.
4: Calcular la probabilidad a posteriori de Δ_t, P_Δ, haciendo uso del árbol de unión.
5: Si $s = VB$, entonces la varianza de P_Δ será $\min\left(\sigma^2_{(qh,d)}, \sigma^2_x\right)$ donde $\sigma^2_{(qh,d)}$ es la varianza histórica calculada para el par (QH, D), y σ^2_x es la varianza calculada de P_Δ. En cualquier otro caso se deja la calculada en P_Δ.
6: Actualizar S_{t-1} con P_{S_t} y Δ_{t-1} con P_Δ. Volver a paso 1.

En el paso 3, para calcular $P(S_t = s|O_{1:t})$ se transforma el problema como sigue haciendo uso de la regla de Bayes,

$$P(S_t = s|O_{1:t}) = \frac{P(S_t = s, O_{1:t})}{P(O_{1:t})}. \tag{4.10}$$

$P(O_{1:t})$ es igual en todos los estados, ya que las observaciones son las mismas en todos los casos, lo que permite eliminar el término $P(O_{1:t})$. Por tanto, maximizar la probabilidad $P(S_t = s|O_{1:t})$ es equivalente a maximizar $P(S_t = s, O_{1:t})$ y para ello se hace uso del algoritmo 3 descrito en el capítulo 3, ya que se quiere calcular la probabilidad de que una variable oculta tenga un estado dada una secuencia de observaciones. Recordar que la variable α_t calculada en el algoritmo representa precisamente la probabilidad conjunta de que se haya producido la secuencia de observaciones $O_{1:t}$ y de que el sensor en tiempo t, S_t, tome el valor s, esto es, $P(S_t = s, O_{1:t})$. Además, para evitar problemas numéricos se toman logaritmos en los cálculos de las probabilidades.

4.2.4. Análisis de los datos

Como se ha indicado, los datos sobre los que se va a trabajar son datos obtenidos del bosque *Andrews*, concretamente de tres estaciones meteorológicas: *Primary*, *Central* y *Upper Lookout*. Cada una de estas estaciones meteorológicas tiene cuatro sensores de temperatura colocados a una altura de 1.5, 2.5, 3.5 y 4.5 metros sobre el suelo, para los que se tienen datos desde 1996 hasta 2008 con una frecuencia cuarto horaria. Como se observa en la figura 4.6, la temperatura tiene una fuerte componente diaria y estacional puesto que, en un día normal, la temperatura sube por la mañana para bajar por la tarde. La componente estacional se puede ver en la diferencia en la temperatura entre

las estaciones. Por ejemplo, es usual que en invierno la temperatura sea más baja que en verano.

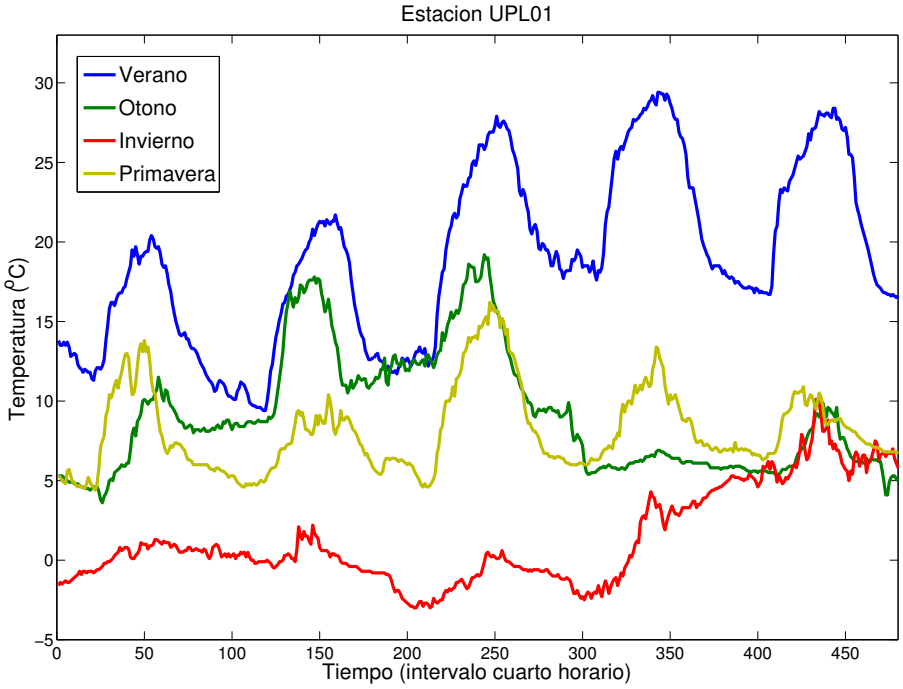
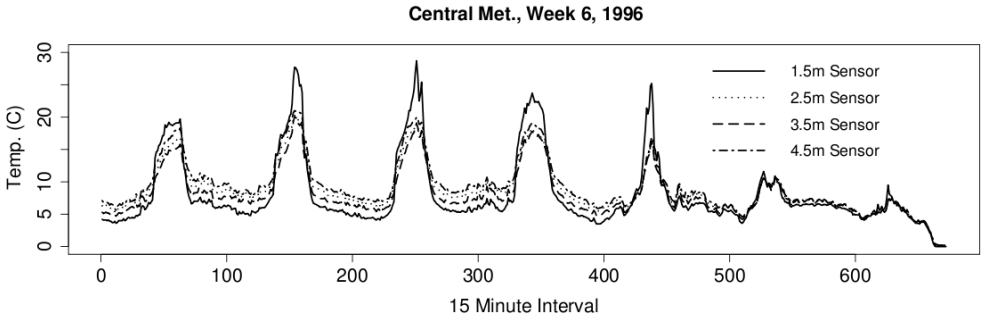


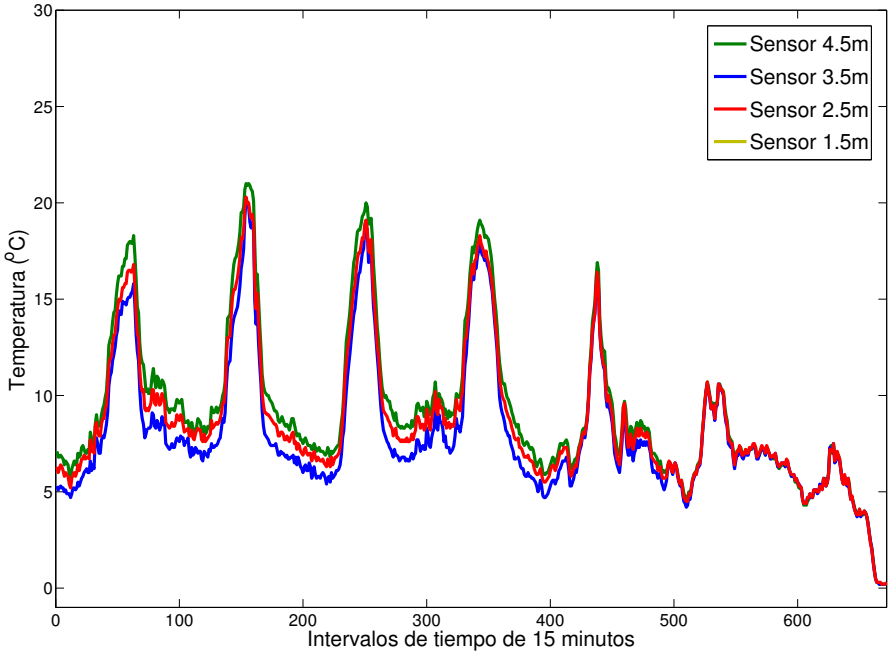
Figura 4.6: Series de temperatura de cinco días para las distintas estaciones del año.

En los datos vienen identificadas tres tipos de anomalías: la ausencia de datos, el sensor cubierto por la nieve y la presencia de datos cuestionables. Por otra parte, se ha observado que los datos no son exactamente los mismos a los usados en el artículo de Dereszynski, como se puede observar en la figura 4.7. En ésta se observa que el sensor de altura 1.5 metros no aparece en los datos de este trabajo (debido a que hay ausencia de datos en ese intervalo), mientras que tiene unos valores anormales en los datos del artículo de Dereszynski. Por este motivo los resultados globales no serán comparables. El objetivo del experimento, por tanto, es ver si nuestra red detecta esos tipos de anomalías.

La ausencia de datos es el fallo del sensor más simple de detectar. En los datos se marca como -9999.999. Es la anomalía más común puesto que representa un 80.0 % de los fallos, pero la menos interesante debido a que no supone una dificultad. En segundo lugar están los datos cuestionables, que son datos de temperatura sobre los que el experto duda de su fiabilidad. Éstos representan un 15.8 % de las anomalías. En último lugar,



(a)



(b)

Figura 4.7: Datos reales de temperatura para el mismo intervalo de tiempo. Figura 4.7(a):datos del artículo de Dereszynski. Figura 4.7(b) datos usados en este trabajo.

pero probablemente los más importantes, están los datos de temperatura en los que el sensor está cubierto por la nieve. Éstos representan un 4.2 % del total de anomalías.

Los dos últimos tipos de anomalías son más complejos que el primero, ya que en ellos se siguen recibiendo datos dentro del rango válido pero no son datos correctos. El primero de

Tabla 4.2: Años de entrenamiento utilizados para cada sensor.

Sensor	**Años**
CEN01	2000-2003
CEN02	1998-2001
CEN03	1998-2001
CEN04	2000-2003
PRI01	2002-2005
PRI02	2005-2008
PRI03	2004-2007
PRI04	1999-2002
UPL01	2005-2008
UPL02	1999-2002
UPL03	1998-2001

estos tipos, los datos cuestionables, no es tan crítico porque no varía mucho con respecto al dato real esperado; simplemente no podemos asegurar su veracidad. Respecto a los datos del sensor cubierto por la nieve, son más críticos puesto que recibimos datos bien medidos pero éstos no se corresponden con la realidad. Por ello es importante detectarlos, para al menos tener conocimiento de que no son datos de temperatura ambiental.

4.2.5. Resultados

El objetivo de los experimentos es etiquetar el estado de sensor en cada instante de tiempo para validar la implementación con los experimentos realizados por Dereszynski en [7]. Como hemos indicado anteriormente los datos sobre los que trabaja no son exactamente los mismos y los resultados no son comparables; por tanto se intentará ver que se seleccionan correctamente los estados de los sensores.

En esta sección se resumirán los resultados obtenidos y se dará una breve explicación de los mismos. En primer lugar se debe escoger los años de entrenamiento para realizar el entrenamiento. En él se obtienen los parámetros necesarios de las distribuciones, es decir, la temperatura media para cada par *(día, cuarto de hora)*, $B_{(d,qh)}$, y la media de las desviaciones de temperatura, $\mu_{(d,qh)}$, y la varianza de las desviaciones de temperatura, $\sigma_{(d,qh)}$, usando para ello las ecuaciones 4.4, 4.6 y 4.7 respectivamente. En la tabla 4.2 se recoge la selección de años de entrenamiento, en la que se han tomado los años con mayor cantidad de datos disponibles.

Tabla 4.3: Resultados obtenidos en la selección de la ponderación de Δ_{t-1} en los distintos sensores. Donde "Acierto" indica el porcentaje de estados del sensor correctamente clasificados.

Sensor	**w**	**Acierto**
CEN01	0.8	99.99 %
CEN02	0.75	99.99 %
CEN03	0.8	99.99 %
CEN04	0.8	99.52 %
PRI01	0.75	100 %
PRI02	0.75	100 %
PRI03	0.75	100 %
PRI04	0.75	100 %
UPL01	0.85	99.99 %
UPL02	0.85	100 %
UPL03	0.85	95.50 %

A continuación se debe ajustar el parámetro libre que tenemos en la red, w, que representa la ponderación de la desviación del instante de tiempo anterior, $w\Delta_{t-1}$. Para realizar este ajuste, se toma un año de los datos de entrenamiento para cada uno de los sensores y se va variando w en un rango de 0.1 a 1 con un paso de 0.05. Una vez obtenidos los resultados de las distintas ejecuciones se toma para cada sensor el w que más haya acertado en la clasificación del estado del sensor en el año seleccionado.

La tabla 4.3 muestra los resultados obtenidos en la selección. Como se observa, todos los w seleccionados poseen valores altos, esto se debe a que durante las ejecuciones el acierto va a aumentando a la par que el parámetro w hasta que se llega a un punto en el que no varía más. Un ejemplo de esto lo podemos ver en la figura 4.8.

Una vez seleccionado el parámetro de ponderación, ya se pueden ejecutar los años de test. A continuación se evaluarán los resultados por separado para cada una de las estaciones meteorológicas en los años de test.

4.2.5.1. *CENTRAL*

La tabla 4.4 muestra las tasas de acierto de los sensores CEN01 (altura 4.5m), CEN02 (altura 3.5m), CEN03 (altura 2.5m) y CEN04 (altura 1.5m) respectivamente en los años de test seleccionados. Las tasas de acierto son el porcentaje de etiquetas acertadas el la clasificación de los sensores. Se observa que, por lo general, las tasas de acierto son muy

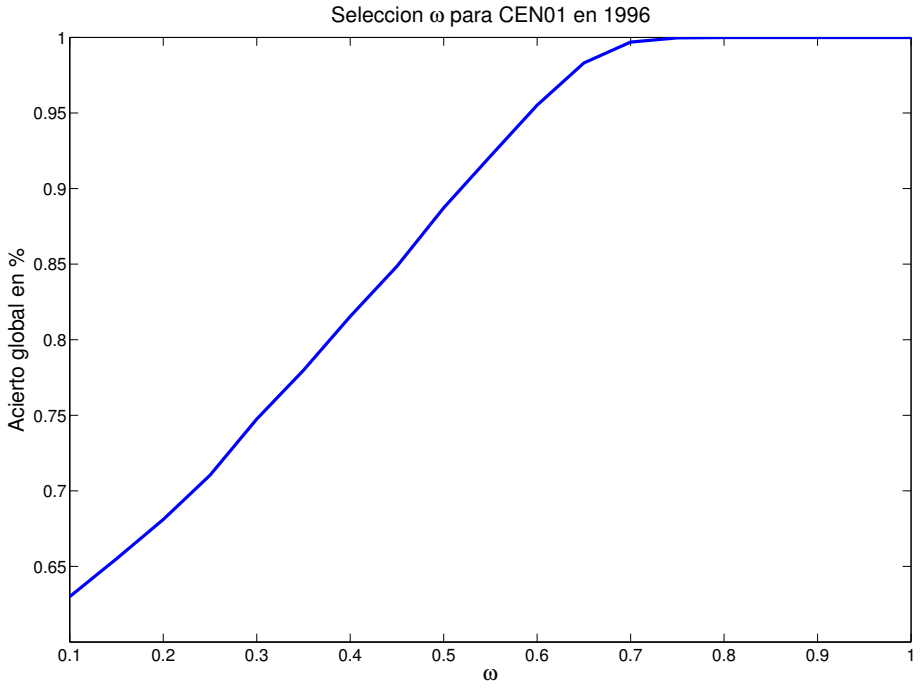


Figura 4.8: Tasa de acierto en función del valor del parámetros w de ponderación de Δ_{t-1} en el sensor CEN01 para el año 1998.

altas. Esto se debe a que el número de datos erróneos en dichos sensores es muy bajo y a que la mayoría de las incidencias existentes son producidas por ausencia de datos.

La mayoría de los fallos cometidos por el modelo al etiquetar el estado del sensor se producen porque el modelo tarda en recuperar la tendencia local de la temperatura tras una ausencia de datos prolongada y, por tanto, la predicción se encuentra desviada de la real. Este efecto queda patente en las matrices de confusión. Por ejemplo, la del caso del sensor CEN01 en el año 2004 representada en la tabla 4.5, donde se observa que parte de los datos que eran medidas buenas son etiquetadas como B. En la figura 4.9 se puede ver que estos fallos se producen cuando tras una ausencia de datos la predicción tiene que recuperar la tendencia de la temperatura.

También hay errores debidos a que el modelo predice los datos cuestionables como datos completamente válidos, y algunos datos válidos como cuestionables, tal y como muestran las matrices de confusión del sensor CEN02 para el año 2008 en la tabla 4.6 y del sensor 2003 para el año 2007 en la tabla 4.7.

Tabla 4.4: Tasas de acierto para los distintos años de test de los sensores de la estación *Central*. Entendiendo por acierto el porcentaje de datos para el que se ha clasificado correctamente el estado del sensor.

Año	**Acierto CEN01**	**Acierto CEN02**	**Acierto CEN03**	**Acierto CEN04**
1997	99.99 %	99.99 %	100 %	99.99 %
2004	99.98 %	99.98 %	99.98 %	99.98 %
2005	100 %	99.99 %	99.99 %	100 %
2006	99.97 %	99.64 %	99.98 %	99.97 %
2007	99.99 %	99.98 %	99.98 %	99.98 %
2008	100 %	99.76 %	99.99 %	97.58 %

Tabla 4.5: Matriz de confusión del sensor CEN01 en 2004.

		Pred			
		VG	**G**	**B**	**VB**
Real	**VG**	99.98 %	0 %	0.02 %	0 %
	G	-	-	-	-
	B	-	-	-	-
	VB	0 %	0 %	0 %	100 %

Tabla 4.6: Matriz de confusión del sensor CEN02 en 2008

		Pred			
		VG	**G**	**B**	**VB**
Real	**VG**	99.76 %	0.22 %	0.01 %	0.01 %
	G	-	-	-	-
	B	-	-	-	-
	VB	0 %	0 %	0 %	100 %

Tabla 4.7: Matriz de confusión del sensor CEN03 en 2007

		Pred			
		VG	**G**	**B**	**VB**
Real	**VG**	99.99 %	0 %	0.01 %	0 %
	G	100 %	0 %	0 %	0 %
	B	-	-	-	-
	VB	0 %	0 %	0 %	100 %

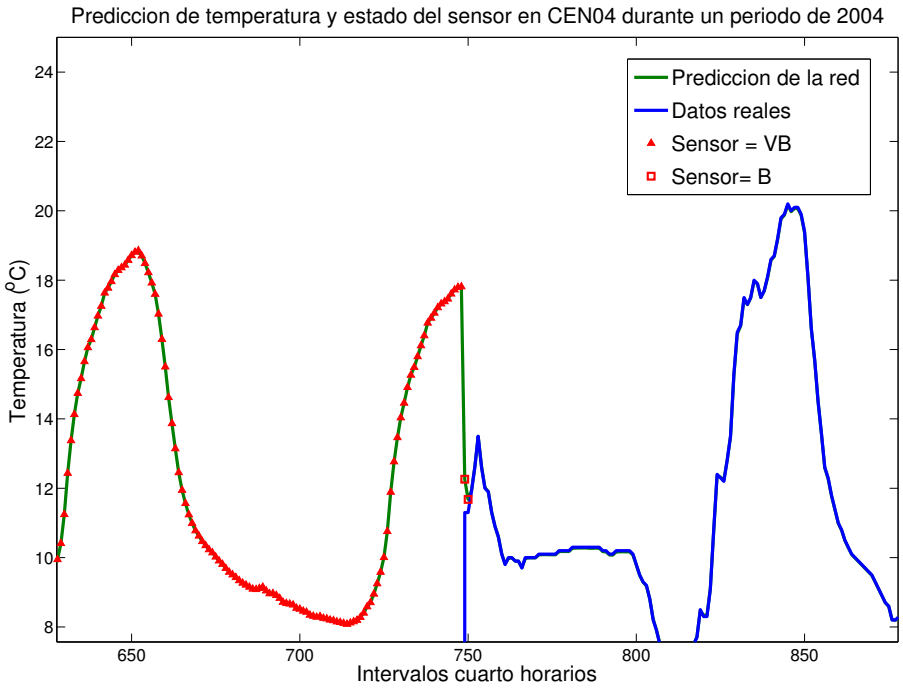


Figura 4.9: Valor de la temperatura real y predicha para el sensor CEN04 en un periodo del año 2004. Ejemplo en el que la predicción no se recupera en dos intervalos de tiempo y se etiqueta el dato como erróneo.

4.2.5.2. *PRIMARY*

La tabla 4.8 muestra las tasas de acierto de los sensores PRI01 (altura 4.5m), PRI02 (altura 3.5m), PRI03 (altura 2.5m) y PRI04 (altura 1.5m) respectivamente en los años de test seleccionados.

En la estación *Primary*, la mayoría de las anomalías son ausencia de datos, aunque encontramos más casos de datos cuestionables que en *Central*. Como efecto, se puede observar en la tabla 4.8 que la tasa de acierto baja significativamente en algún año, en especial el año 1998 para el sensor PRI04.

Al igual que en la estación *Central*, por lo general las tasas de acierto son muy altas debido a la gran cantidad de datos con estado del sensor VG y muy pocos datos con el resto de etiquetas, entre los cuales la mayoría tiene la etiqueta de ausencia de datos. Esto caso se puede observar en la matriz de confusión de sensor PRI03 en el año 2008 representada en la tabla 4.9.

Tabla 4.8: Tasas de acierto para los distintos años de test de los sensores de la estación *Primary*. Entendiendo por acierto el porcentaje de datos para el que se ha clasificado correctamente el estado del sensor.

Año	**Acierto PRI01**	**Acierto PRI02**	**Acierto PRI03**	**Acierto PRI04**
1997	100 %	100 %	100 %	100 %
1998	99.74 %	99.73 %	99.72 %	43.85 %
1999	99.99 %	99.99 %	99.98 %	-
2001	99.97 %	99.97 %	99.97 %	-
2002	-	73.48 %	73.48 %	-
2003	-	100 %	99.98 %	99.98 %
2004	-	99.98 %	-	99.99 %
2005	-	-	-	99.99 %
2006	99.97 %	-	-	99.93 %
2007	99.98 %	-	-	99.98 %
2008	99.97 %	-	94.47 %	99.98 %

Tabla 4.9: Matriz de confusión del sensor PRI03 en 2008

		Pred			
		VG	**G**	**B**	**VB**
Real	**VG**	99.95 %	0.03 %	0.02 %	0 %
	G	99.90 %	0 %	0.10 %	0 %
	B	-	-	-	-
	VB	0 %	0 %	0 %	100 %

Tabla 4.10: Matriz de confusión del sensor PRI04 en 1998

		Pred			
		VG	**G**	**B**	**VB**
Real	**VG**	99.97 %	0 %	0.03 %	0 %
	G	99.69 %	0.27 %	0.04 %	0 %
	B	-	-	-	-
	VB	0 %	0 %	0 %	100 %

Tabla 4.11: Matriz de confusión del sensor PRI04 en 1998 con los datos en bruto.

		Pred			
		VG	**G**	**B**	**VB**
Real	**VG**	14591	0	4	0
	G	19663	53	8	0
	B	-	-	-	-
	VB	0	0	0	721

Sin embargo, para el sensor PRI04 en el año 1998 se obtiene una tasa de acierto muy baja, como se observa en la tabla 4.8. Se debe a que los datos cuestionables son etiquetados como datos válidos, y en algunos pocos casos como inválidos. Al haber un alto porcentaje de datos cuestionables en este año, se produce mucho más error que en otros años. Esto se puede ver en su matriz de confusión, representada en la tabla 4.10 y en la tabla 4.11. Esta última muestra la matriz de confusión sin porcentajes, con número de datos clasificados en cada zona. Se observa que efectivamente se predicen muchos más datos cuestionables que correctos. El caso de los sensor PRI02 y PRI03 en el año 2002 es idéntico al ilustrado en este párrafo.

4.2.5.3. *Upper Lookout*

La tabla 4.12 muestra las tasas de acierto de los sensores UPL01 (altura 4.5m), UPL02 (altura 3.5m) y UPL03 (altura 2.5m) respectivamente en los años de test seleccionados.

En la estación *Upper Lookout*, la mayoría de las anomalías son ausencia de datos. En este caso no hay datos cuestionables y sí hay un caso en el que el sensor está cubierto por la nieve.

Al igual que en los casos anteriores, como se puede observar en la tabla 4.12, las tasas de acierto son muy altas, debido a que apenas hay anomalías distintas a la ausencia de datos. Los fallos en la predicción del estado del sensor en las anomalías con ausencia de datos se producen, como se comenta en el apartado de resultados de la estación *Central*, por los instantes de tiempo que tarda la predicción en recoger la tendencia local y poder clasificar correctamente el estado del sensor. Por ello, algunos de los datos en los que el sensor está en el estado VG se etiquetan como G o B, como se puede observar en la matriz de confusión del sensor UPL01 en el año 1999, representada en la tabla 4.13.

Tabla 4.12: Tasas de acierto para los distintos años de test de los sensores de la estación *Upper Lookout*. Entendiendo por acierto el porcentaje de datos para el que se ha clasificado correctamente el estado del sensor.

Año	**Acierto UPL01**	**Acierto UPL02**	**Acierto UPL03**
1997	99.98 %	99.97 %	99.98 %
1998	99.94 %	100 %	-
1999	99.85 %	-	-
2001	99.86 %	-	-
2002	99.74 %	-	99.98 %
2003	99.89 %	99.98 %	100 %
2004	99.98 %	99.98 %	99.98 %
2005	-	99.98 %	99.98 %
2006	-	100 %	99.99 %
2007	-	99.98 %	99.98 %
2008	-	97.04 %	99.99 %

Tabla 4.13: Matriz de confusión del sensor UPL01 en 1999.

		Pred			
		VG	**G**	**B**	**VB**
Real	**VG**	99.85 %	0.12 %	0.03 %	0 %
	G	-	-	-	-
	B	-	-	-	-
	VB	0 %	0 %	0 %	100 %

Tabla 4.14: Matriz de confusión del sensor UPL02 en 2008

		Pred			
		VG	**G**	**B**	**VB**
Real	**VG**	99.98 %	0 %	0.02 %	0 %
	G	-	-	-	-
	B	100 %	0 %	0 %	0 %
	VB	0 %	0 %	0 %	100 %

En el caso de que se presenten datos detectados por los expertos como cubiertos por la nieve, como en el sensor UPL02 en el año 2008, el error del modelo se produce porque al ser las temperaturas habituales bastante bajas, cuando queda el sensor sepultado por la nieve la temperatura puede no sufrir un cambio brusco o una desviación de su temperatura habitual. Esto se puede observar en la matriz de confusión del sensor UPL02 en el año 2008, representada en la tabla 4.14 y en la figura 4.10, que representa el caso de un sensor cubierto por la nieve.

4.2.6. Conclusiones

Como se ha visto en la sección de resultados, el modelo propuesto no es capaz de detectar las anomalías más complicadas e interesantes. Detecta correctamente el buen funcionamiento del sensor, pero en el momento de detectar los datos considerados como cuestionables, o los datos obtenidos de un sensor cubierto por la nieve, el modelo los etiqueta como datos correctos, como se puede observar en la tabla 4.14 en la que se tiene el caso de un sensor cubierto por la nieve.

En éste caso, se ha comentado anteriormente que el principal problema que encuentra el modelo, es que las temperaturas registradas bajo la nieve son similares o incluso un poco más altas que las obtenidas cuando el sensor funcionaba correctamente. Podemos ver un ejemplo en la figura 4.10.

En el caso de los datos cuestionables, el problema viene de que apenas hay variación entre la temperatura esperada y la temperatura registrada. Por ello la red es incapaz de detectar dicha anomalía, es decir, no se sale de lo esperable.

A pesar de que el modelo no ha sido capaz de clasificar correctamente las anomalías más complicadas si que ha sido capaz clasificar correctamente los datos bien medidos, lo que indica que el modelo predictivo de la red ha funcionado como se esperaba. Esto

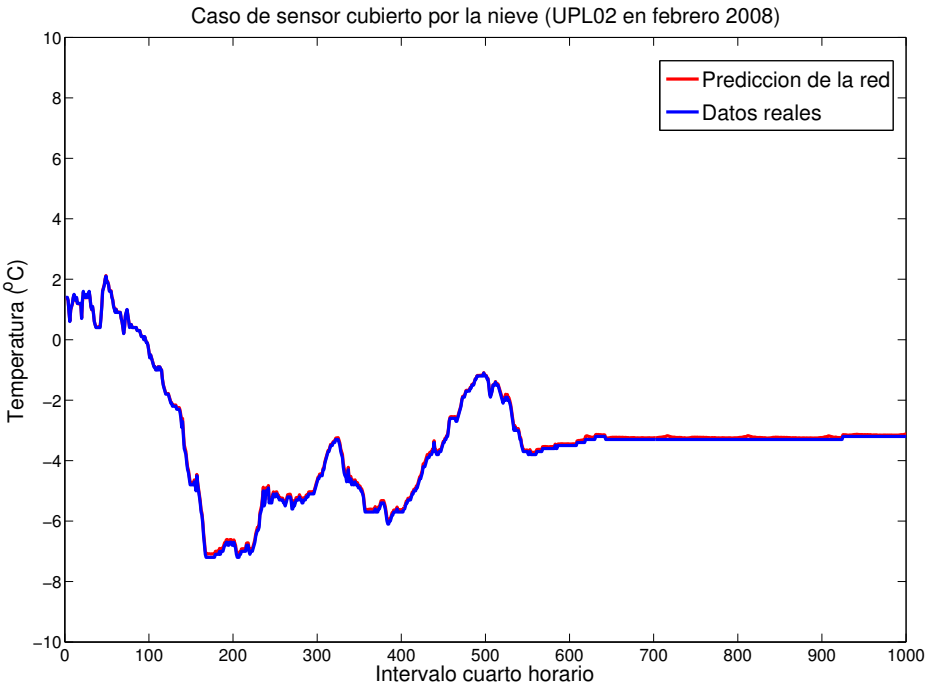


Figura 4.10: Situación en la que el sensor está cubierto por la nieve en el sensor UPL02 en el año 2008.

nos dice que la implementación de la teoría de las gaussianas condicionales aplicadas a las redes bayesianas híbridas y dinámicas funciona.

4.3. Radiación solar

En esta sección se exponen los experimentos realizados para la aplicación de la teoría del trabajo a cuestiones puramente predictivas. Para ello se han utilizado datos de medidas reales de radiación solar tomados en la península así como la librería de código Matlab *Bayes Net Toolbox* creada por Kevin P. Murphy [11] y disponible en http://bnt.googlecode.com.

4.3.1. Problema

El objetivo es mejorar la predicción de radiación solar que incide sobre la Tierra otorgada por el ECMWF (*European Center for Medium-Range Weather Forecasts* [1]). Para conseguirlo proponemos aplicar una red bayesiana dinámica para construir un modelo

[1] http://www.ecmwf.int

predictivo e intentar capturar no sólo la estacionalidad sino también la dependencia del instante anterior.

4.3.2. Análisis de los datos

Los datos con los que se va a trabajar en este apartado son:

- Medidas reales de radiación horarias tomadas en el aeropuerto de Granada entre 2005 y 2010.
- Predicción de radiación agregada trihoraria realizada por el ECMWF para los años 2009 y 2010 en las coordenadas más cercanas al aeropuerto de Granada.
- Predicción de nubosidad trihoraria del ECMWF para los años 2009 y 2010 en el mismo punto que la predicción de radiación.

Debido a los pocos datos del ECMWF que se poseen, sólo podremos trabajar con las medidas reales tomadas en 2009 y 2010. El otro inconveniente es que la predicción del ECMWF viene acumulada trihorariamente en contraposición a los datos horarios de las medidas reales.

4.3.3. Modelo

Para predecir la radiación solar se trabaja con las gaussianas condicionales, propuestas por Lauritzen [10] y descritas en el capítulo 2, sobre una red bayesiana dinámica. La separación entre instantes de tiempo será de 1 hora, que es el intervalo de refresco de las medidas reales en los datos. Por ello la red contendrá dos variables discretas que representan el instante de tiempo que nos encontramos, (H, D), donde:

1. $H = 1, ..., 24$ representa, para un día, la hora en la que nos encontramos.
2. $D = 1, ..., 365$ representa el día del año en el que nos encontramos.

Haciendo uso de dicha red bayesiana, se quiere realizar una predicción de la radiación solar que mejore la dada por el ECMWF. Para ello se asume que la radiación depende de un valor base, B, y una corrección sobre el mismo, Δ. La distribución de la radiación viene dada por:

$$R \sim N\left(B_{(h,d)} + \Delta_t, \sigma_R^2\right). \tag{4.11}$$

Para el valor de B vamos a realizar dos aproximaciones:

- Usar la predicción de radiación teórica, *clear sky*, ponderada por un factor teniendo en cuenta la predicción de nubosidad para esa hora.
- Usar directamente la predicción de radiación del ECMWF.

4.3.3.1. *Clear sky* ponderado

La predicción de radiación de *clear sky* es una predicción teórica suponiendo que no hay nubosidad. Se puede profundizar más en su definición en el artículo de Richard Bird [3] y en el libro de Iqbal [9].

En el modelo propuesto se toma la predicción de radiación *clear sky* para el par *(día,hora)* en el que se está operando, R_{csky}, y se realiza un ajuste teniendo en cuenta la nubosidad predicha por el ECMWF para ese día a esa hora, ρ_{nub}. Se propone que:

$$B = R_{csky} \times \rho_{nub}, \tag{4.12}$$

donde ρ_{nub} tendría un valor de r_{real}/R_{csky}, donde r_{real} es el valor real de la radiación. Esta división debería valer 1 si no hay nubes e ir disminuyendo según aumente la nubosidad. Por ello se propone tomar $\hat{\rho}_t = 1 - \delta n_t$, donde $\hat{\rho}_t$ es el factor para el instante de tiempo t, n_t es la nubosidad en el instante de tiempo t y δ es un valor entre 0 y 1 que se ajustará durante el entrenamiento con un conjunto de validación. La fórmula para calcular el valor base para un par *(día,hora)*

$$B_{(h,d)} = R_{csky}\,(h,d) \times \hat{\rho}_t, \tag{4.13}$$

Como se observa en la figura 4.11 la radiación tiene una alta estacionalidad, por lo que se propone tomar un valor de δ para cada estación de año.

Una vez se tiene definido el valor base se define la desviación Δ_t sobre éste, que se puede interpretar como una corrección teniendo en cuenta la tendencia local de la radiación. Por ello, Δ_t se modela como un proceso de *Markov* de primer orden con el par (h, d) como entradas observadas, donde h representa la hora y d el día. Su distribución viene dada por:

$$\Delta_t \sim N\left((1-w)\,\mu_{(h,d)} + w\Delta_{t-1}, \sigma_{(h,d)}\right). \tag{4.14}$$

A diferencia de los experimentos con temperatura, la media viene determinada por una ponderación convexa de media histórica de la desviación para ese par *(día,hora)*, $\mu_{(h,d)}$, y la desviación en el instante anterior, Δ_{t-1}. Esto se debe a que durante los experimentos preliminares se observó que este modelo era más estable ante la variación de w como se puede ver en la figura 4.12. Por otra parte, la varianza viene dada por la varianza histórica de las desviaciones para el par *(día,hora)*, al igual que en el caso de la temperatura. De esta forma capturamos la componente estacionaria de la radiación así como la componente local.

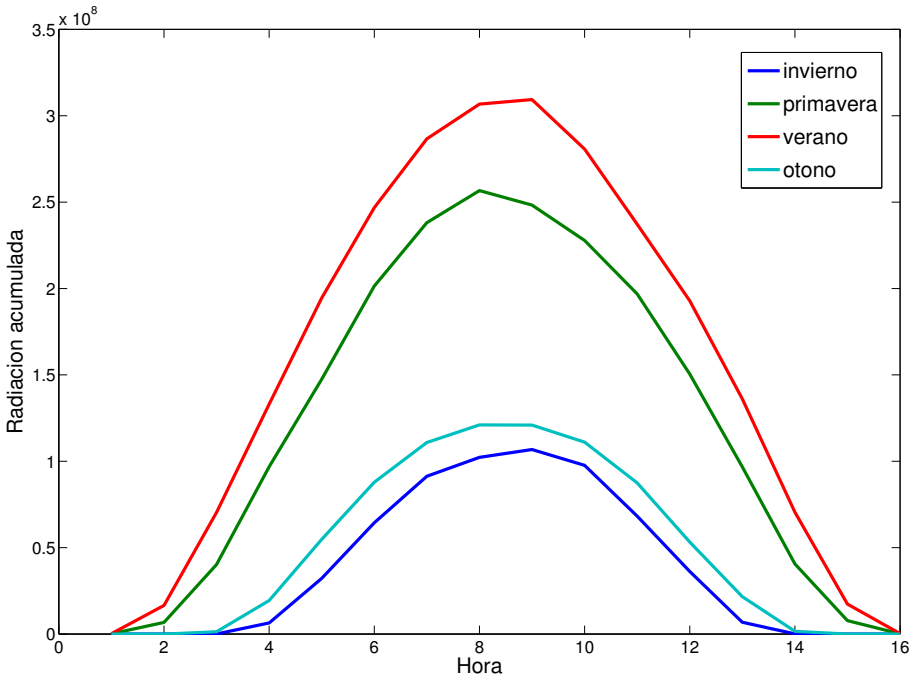


Figura 4.11: Radiación medida total por horas para las distintas estaciones del año.

Al igual que se hizo en los experimentos de temperatura y debido a los pocos datos disponibles, recordar que sólo podemos trabajar con dos años de datos, 2009 y 2010, la media y la varianza históricas se calculan usando una ventana de 31 días y los dos años de datos de la siguiente forma:

$$\mu_{(h,d)} = \frac{1}{Y\,(2M+1)} \sum_{y,u} \Delta\,(y, d+u, h)\,. \tag{4.15}$$

$$\sigma^2_{(h,d)} = \frac{1}{Y\,(2M+1)} \sum_{y,u} \left(\Delta\,(y, d+u, h) - \mu_{(h,d)}\right)^2. \tag{4.16}$$

donde $\Delta\,(y, d, h)$ se calcula como:

$$\Delta\,(y, d, h) = r_{real}\,(y, d, h) - B_{(h,d)} \tag{4.17}$$

Para poder realizar este modelo se asume que la nubosidad dada por el ECMWF para la hora 6 será la nubosidad para las horas 6,7 y 8, estando la hora en formato solar. De esta forma podemos aplicar el modelo de forma horaria.

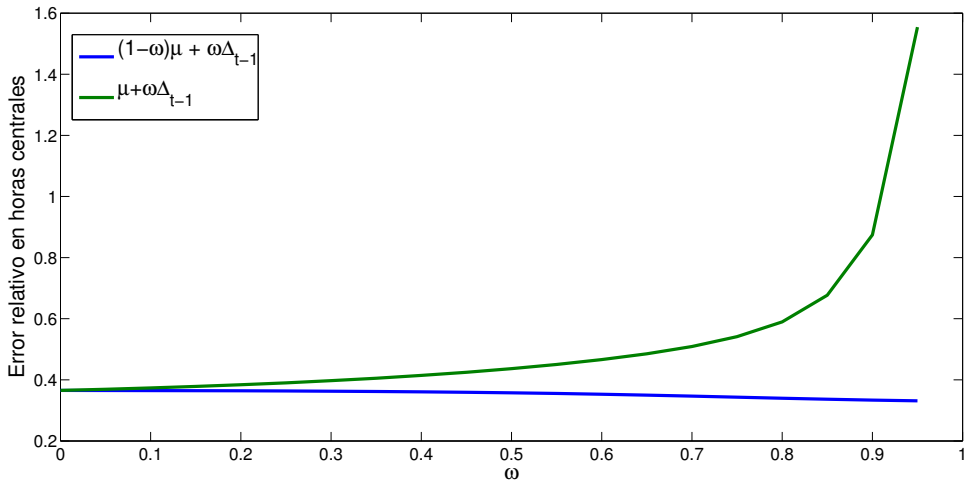


Figura 4.12: Error relativo en horas centrales para distintos valores de w en las distintas aproximaciones de la desviación Δ_t para el modelo de *clear sky* ponderado.

4.3.3.2. ECMWF

Esta aproximación podemos tratarla más que como una predicción de radiación como una corrección de la predicción previa dada por el ECMWF. En este caso el valor base será la predicción de radiación del ECMWF para el par *(día,hora)* en el que se encuentre la red. Será una predicción externa y ajena al modelo.

En este caso, Δ_t se puede interpretar como el error cometido por la predicción. Al igual que en casos anteriores se modela como un proceso de *Markov* de primer orden con el par (h, d) como entradas observadas. Su distribución será igual que en el caso de la ecuación 4.14. En este caso la media vendrá determinada por la media del error histórico para el par *(día,hora)*, $\mu_{(h,d)}$, y una ponderación del error estimado en el instante anterior. En el caso de la varianza, vendrá determinada por la varianza histórica del error cometido para el par *(día,hora)*.

Como los datos del ECMWF están acumulados trihorariamente, se acumulan las medidas de radiación en los mismos intervalos para que se muevan en el mismo rango y poder calcular de esta manera los errores históricos.

La media y la varianza históricas se calculan durante el entrenamiento usando una ventana de 10 días y los dos años de datos disponibles de la misma forma a la encontrada en las ecuaciones 4.15 y 4.16.

Con estos conceptos se crea la red bayesiana dinámica usada para predecir, que en ambos casos tendrá la estructura mostrada en la figura 4.13.

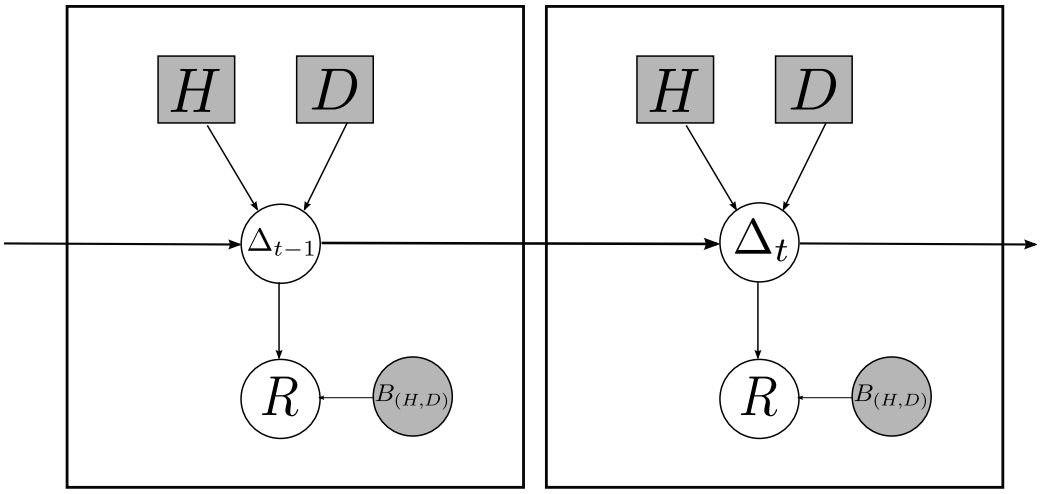


Figura 4.13: Estructura de la red bayesiana dinámica en el modelo para radiación solar.

4.3.4. Inferencia

La inferencia en la red se realiza usando el árbol de unión, aplicado a las CG's por Lauritzen en [10] y visto en el capítulo 2. En primer lugar se deberá crear el árbol de unión a partir del grafo formado por la ventana de tiempo actual y las variables de la ventana anterior que tengan influencia sobre la actual, como se vió en el capítulo 3. En la figura 4.14 podemos ver el grafo a partir del cual se creará el árbol de unión. El grafo de unión generado se puede ver en la figura 4.15. Una vez se tiene el árbol de unión se propone una secuencia de acciones descritas en el algoritmo 5. Con ellas se llevará a cabo la inferencia en la red.

Algoritmo 5 Algoritmo usado para la inferencia de la red en la predicción de radiación.

1: Introducir el valor de las variables observadas: H, D, $B_{(h,d)}$.
2: **si** Es de noche **entonces**
3: La predicción de radiación, R, será 0.
4: **si no**
5: Computar la distribución de Δ_t para el instante (H, D).
6: Calcular la predicción de radiación como $B_{(h,d)} + \Delta_t$.
7: Actualizar Δ_{t-1}.
8: **fin si**
9: Volver a punto 1.

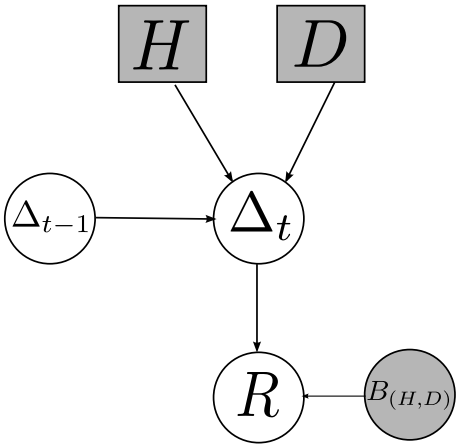


Figura 4.14: Estructura de la red bayesiana dinámica en el modelo para radiación solar.

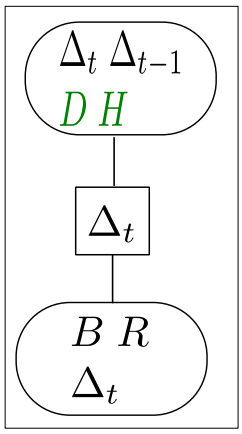


Figura 4.15: Árbol de unión generado a partir del grafo de la figura 4.14.

4.3.5. Medidas de calidad

Para comprobar la bondad del modelo se va a utilizar el error relativo en las horas centrales del día, entre las 10 de la mañana y las 15 de la tarde en hora solar.

$$err = \frac{|\hat{R}(h) - R(h)|}{R(h)}, \tag{4.18}$$

donde $\hat{R}(h)$ es la predicción para la hora h, y $R(h)$ es la radiación real medida para la hora h. De esta forma se tiene un error relativo para cada hora del día. Luego, se hará uso de la media de esos errores para obtener errores medios en rangos de tiempo más amplios, como estaciones del año.

Tabla 4.15: Valores seleccionados para δ.

Estación	δ
Invierno	0.8
Primavera	0.7
Verano	0.3
Otoño	0.7

Se han escogido estas horas porque son las horas con más radiación y por tanto, las más interesantes en predecir correctamente. Además, no se tiene en cuenta las horas con baja radiación porque en ellas el valor de $R(h)$ es muy pequeño y por tanto, usando la fórmula 4.18, obtendríamos un valor muy alto para el error relativo.

4.3.6. Resultados

En esta sección se describirán los resultados obtenidos para las dos aproximaciones propuestas.

4.3.6.1. *Clear sky* ponderado

Como año de test en los experimentos vamos a utilizar el 2009. Debido a la escasez de datos, tan sólo se puede trabajar con dos años de datos, la selección de δ se realiza sobre los resultados de test, de forma que se ejecuta el modelo para todas las combinaciones posibles, variando δ entre 0 y 1 con un paso de 0.1 y seleccionando la combinación que mejor resultados ofrece. Recordar que δ es el factor de la nubosidad que tenemos en cuenta para calcular el factor de corrección del *clear sky*. Los valores que ofrecen mejores resultados para cada estación se pueden ver en la tabla 4.15. En cuanto se dispongan de más datos, una de las tareas a realizar es comprobar la bondad de esta selección en el nuevo año de datos.

Respecto al valor de w, como se puede observar en la figura 4.12 obtenida tras la ejecución sobre un año de datos, cuanto mayor es su valor mejores resultados se obtienen. Por tanto se selecciona el valor 1, de forma que se ignora el valor de la media histórica y sólo se tiene en cuenta la deviación previa de la radiación.

Una vez seleccionados los mejores resultados, se han comparado las predicciones obtenidas con las del ECMWF y un modelo de persistencia, consistente en que la predicción para cierta hora es la observación para la misma hora en el día anterior. La tabla 4.16

Tabla 4.16: Media del error relativo en horas centrales del ECMWF, la red bayesiana (*clear sky* más nubosidad), la red bayesiana con base la predicción del centro europeo y la persistencia.

Error relativo	**ECMWF**	**Red bayesiana *Clear sky***	**Red bayesiana ECMWF**	**Persistencia**
Anual	29.26 %	**24.76 %**	25.82 %	43.26 %
Invierno	61.04 %	**40.31 %**	53.88 %	83.52 %
Primavera	27.19 %	24.97 %	**24.22 %**	36.70 %
Verano	7.67 %	8.67 %	**6.67 %**	10.26 %
Otoño	21.75 %	25.44 %	**19.07 %**	43.26 %

contiene las medías del error para los tres modelos en los siguientes periodos: anual, invierno, verano, primavera y otoño. Como se puede observar en ella, el modelo propuesto comete menos error en media durante las estaciones de invierno y primavera, siendo en la estación de invierno en la que se produce la mejora más significativa. En el caso del verano y del otoño, el modelo propuesto mejora significativamente a la persistencia pero no es capaz de alcanzar la tasa de error del ECMWF. Se pueden observar estos mismos resultados de una manera más gráfica en el diagrama de barras de la figura 4.17.

4.3.6.2. ECMWF

En el caso del modelo que usa como valor base la predicción del ECMWF también se va a usar el año 2009 para realizar los test. De esta forma se podrán comparar los resultados de ambas aproximaciones. Para este modelo no es necesario seleccionar ningún parámetro de la nubosidad, pero sí que hay que seleccionar el w adecuado, ya que el modelo es diferente y por tanto no podemos usar el resultado obtenido para el modelo *clear sky* ponderado. En este caso el valor de w que mejores resultados otorga es el 0.05, es decir, dándole todo el peso al error histórico y prácticamente ignorando el error del instante anterior. Este resultado se puede observar en la figura 4.16.

Una vez seleccionado el w adecuado, se analizan los errores obtenidos. En la tabla 4.16 se muestran las medias del error para todos los modelos para los periodos: anual, invierno, verano, primavera y otoño. Como se puede observar en ella, aunque el error global es un poco más alto que en el modelo *clear sky* ponderado, en este caso mejoramos en todas las estaciones el error de predicción del ECMWF, siendo únicamente mejor el modelo *clear sky* ponderado en el caso del invierno. Estos resultados eran esperables, ya que se realiza una corrección sobre la predicción teniendo en cuenta el error histórico para cada par *(día,hora)*. También se pueden observar estos resultados de forma gráfica

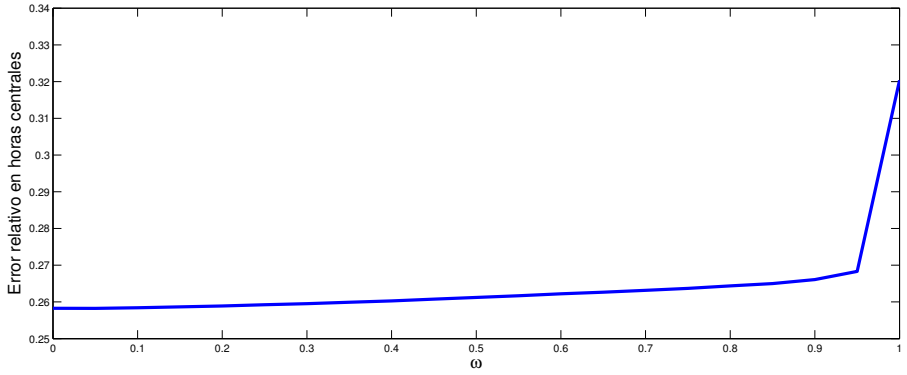


Figura 4.16: Error relativo en horas centrales para distintos valores de w para el modelo con base la predicción del ECMWF.

en el diagrama de barras de la figura 4.17.

4.3.7. Conclusiones

Como se ha visto en la sección de resultados ambos modelos propuestos mejoran la predicción dada por el ECMWF. En el caso de la aproximación que usa como base la estimación *clear sky*, se observa que es muy superior en la estación de invierno al resto de predicciones. En el caso del modelo que usa como base la predicción del ECMWF, se ve que mejora la predicción del ECMWF en todas las estaciones. Esto indica que el modelo propuesto es válido para mejorar una predicción externa.

Aún así, estos resultados no se pueden considerar definitivos puesto que debido a la escasez de datos se ha realizado la selección de parámetros sobre el año de test. No obstante uno de los pruebas a realizar como continuación de este trabajo es comprobar el comportamiento de este modelo en el año 2011, con el fin de validar los resultados obtenidos.

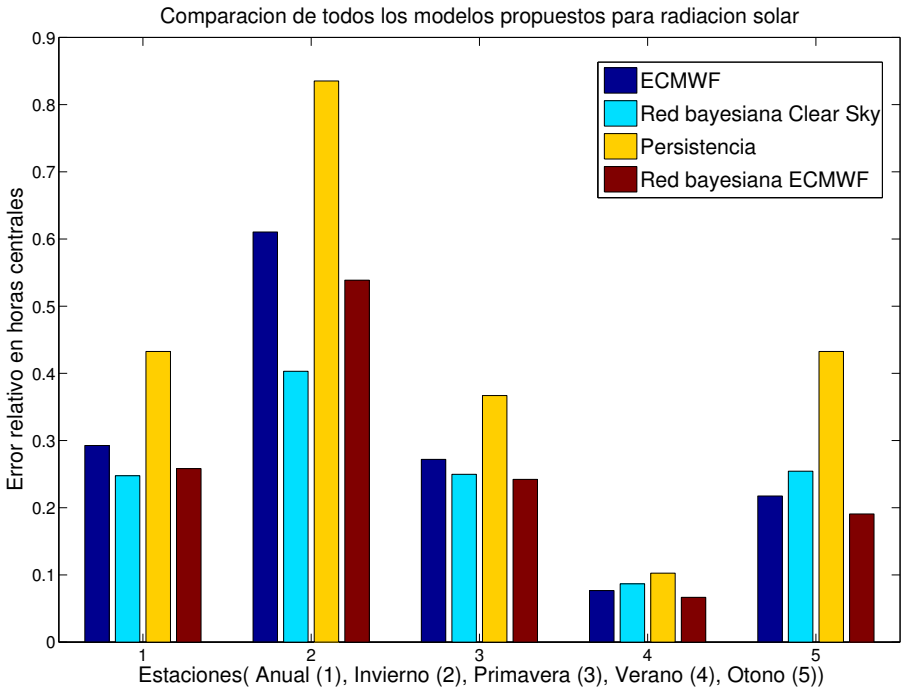


Figura 4.17: Diagrama de barras representando los errores del centro europeo, la red bayesiana (*clear sky* más nubosidad), la persistencia y la red bayesiana, con valor base el centro europeo.

Capítulo 5

Discusión y conclusiones

A lo largo de este trabajo se han introducido brevemente las redes bayesianas, así como la teoría de las gaussianas condicionales propuesta por Lauritzen en [10] para realizar inferencia exacta sobre redes bayesianas híbridas. En base a esta teoría se ha definido su uso en las redes bayesianas híbridas dinámicas para aplicarlas posteriormente a dos problemas.

El primer problema abordado consiste en la clasificación del estado de un sensor de temperatura a partir de la observación que recibimos de él. La intención inicial de estos experimentos era replicar los experimentos realizados por Dereszynski en [7] y conseguir realizar una validación de la teoría introducida y su implementación. Durante la realización de dichos experimentos se observó que los datos que se utilizan en este trabajo difieren de los datos utilizados por Dereszynski, por lo que los resultados no serían comparables. Aún así se han evaluado los resultados en la tarea de clasificar el estado del sensor de temperatura, observando que, a pesar de los altos porcentajes de acierto en la clasificación, el modelo no detectaba correctamente los datos cubiertos por la nieve, así como los datos cuestionables.

El segundo problema abordado es la mejora de la predicción de radiación solar estimada por el ECMWF mediante una red bayesiana dinámica híbrida basada en las condicionales gaussianas. En este caso se llevan a cabo dos aproximaciones, la primera usando como valor base de radiación el valor teórico de radiación sin nubes ponderado por la nubosidad. La segunda aproximación utiliza como valor base la propia predicción del ECMWF, de manera que la red realiza una corrección sobre el error histórico cometido por el ECMWF. En ambos modelos se han obtenido buenos resultados, llegando a mejorar la predicción del ECMWF en todas las estaciones con la segunda de ellas.

Bibliografía

[1] L. E. Baum and J. A. Eagon. An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bulletin of the American Mathematical Society*, 73(3):360–363, 1967.

[2] L. E. Baum and G. R. Sell. Growth functions for transformations on manifolds. *Pacific J. Math*, 27(2):221–227, 1968.

[3] R. E. Bird and R. L. Hulstrom. Simplified clear sky model for direct and diffuse insolation on horizontal surfaces. Technical Report SERI/TR-642-761, Golden, CO: Solar Energy Research Institute, 1981.

[4] G. F. Cooper. The computational complexity of probabilistic inference using bayesian belief networks. *Artif. Intell.*, 42(2-3):393–405, 1990.

[5] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Statistics for Engineering and Information Science. Springer, 1999.

[6] C. Daly and W. McKee. Meteorological data from benchmark stations at the andrews experimental forest. long-term ecological research. *Forest Science Data Bank, Corvallis, OR.[Database]*, 2011.

[7] E. W. Dereszynski. Probabilistic models for anomaly detection in remote sensor data streams. In *23rd Conference on Uncertainty in Artificial Intelligence*, 2007.

[8] R.C. Griffiths and S. Tavare. Ancestral inference in population genetics. *Statistical Science*, 9:307–307, 1994.

[9] Muhammad Iqbal. *An introduction to solar radiation*. Academic Press, 1983.

[10] S. L. Lauritzen. Propagation of probabilities, means, and variances in mixed graphical association models. *Journal of the American Statistical Association*, 87(420):1098–1108, 1992.

[11] K. P. Murphy. The bayes net toolbox for matlab. *Computing Science and Statistics*, 33, 2001.

[12] R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, University of Toronto, 1993.

[13] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of IEEE*, volume 77, pages 257–286, 1989.

[14] T. A. Stephenson. An Introduction to Bayesian Network Theory and Usage. Technical report, Dalle Molle Institute for Perceptual Artificial Intelligence, 2000.

[15] R. Sterritt, A. H. Marshall, C. M. Shapcott, and S. I. McClean. Exploring dynamic bayesian belief networks for intelligent fault management systems. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 3646–3652, 2000.

Índice de tablas

Índice de figuras